



# **Intelligent Laser Scanning for Computer Aided Manufacture**

**Alistair John Denby**

**October 2006**

**A thesis submitted in partial fulfillment of the  
requirements of Nottingham Trent University  
for the degree of Doctor of Philosophy**

**School of Computing and Informatics  
The Nottingham Trent University  
in collaboration with  
Axiomatic Technology (UK) Ltd.**



The work described in this Thesis is the Author's own, unless otherwise stated, and it is, as far as he is aware, original.

This thesis has been supplied on condition that anyone who consults it is understood to recognize that its copyright rests with the Author and that no information derived from it may be published without prior written consent.

## Abstract

Reverse engineering requires the acquisition of large amounts of data describing the surface of an object, sufficient to replicate that object accurately using appropriate fabrication techniques. This is important within a wide range of commercial and scientific fields where CAD models may be unavailable for parts that must be duplicated or modified, or where a physical model is used as a prototype. The three-dimensional digitisation of objects is an essential first step in reverse engineering. Optical triangulation laser sensors are one of the most popular and common non-contact methods used in the data acquisition process today. They provide the means for high resolution scanning of complex objects. Multiple scans of the object are usually required to capture the full 3D profile of the object. A number of factors, including scan resolution, system optics and the precision of the mechanical parts comprising the system may affect the accuracy of the process.

A single perspective optical triangulation sensor provides an inexpensive method for the acquisition of 3D range image data. However, there are often locations within each scan where data is seriously flawed because the data acquisition process is subject to distortions. Such distortions are often associated with edges in the object, where regions of high curvature (relative to the incident angle of the sensor) cause occlusions and secondary reflections of the laser beam, resulting in false height readings. Abrupt changes in surface reflectance or texture can also have similar effects. Previous work has determined that the orientation of the scan head with respect to the edges of the object is a major factor in the degree of such distortions. Combining multiple range images using compensation algorithms has reduced the level of distortion in the integrated data set; however capturing the number of necessary repetitions of the entire scan is very time-consuming.

A development platform has been established to investigate how data distortions may be reduced by the application of image analysis techniques in planning the scan process. By using information on edge location and orientation recovered from a digital camera image, partial scans may then be performed for each determined orientation of the scanner, thereby avoiding much redundant coverage of the entire scan area.

Vectorisation algorithms, based on known edge detection techniques, have been developed to determine the position of vectors corresponding to the discovered edges. Further algorithms have been developed to process these vectors into 'scan regions' corresponding to each particular scanner orientation. When the object is scanned at the orientation corresponding to the scan region the distortions are likely to be much reduced.

Some features of the object geometry, such as small holes or internal corners present a particular problem where a number of scan regions representing different scan orientations overlap. Because of the nature of the scanner such regions are liable to show some level of distortion for all laser orientations. However, these locations can be identified from the camera image and the user alerted to the presence of unreliable data.

Calibration methods relating the image and scan space have been shown to be susceptible to errors caused by optical effects from the camera, such as lens barrel distortion and errors due to parallax. Algorithms have been developed to compensate for these effects and combine the data from a number of partial scans in order to provide a single integrated point cloud.

## Acknowledgements

There are many people I need to thank for their help in completing this project. Firstly many thanks are due to my supervisors – Professor Nasser Sherkat, Dr. Janet Poliakoff and Dr. Caroline Langensiepen (for support above and beyond the call of duty). Throughout my thesis-writing period, they have provided encouragement, sound advice, good teaching and many good ideas. I could not have had better mentors for my PhD, and without their assorted common-sense, knowledge, perceptiveness (and the occasional whip-cracking) I would never have finished.

Thank-you to my examiners, Dr. Peter Plassman (University of Glamorgan) and Dr. Heather Powell (internal examiner) for managing to read the whole thesis so thoroughly, and for the surprisingly enjoyable Viva.

Thanks are also due to the folks at Axiomatic Technology (especially David Moore) for their support and the rest of the academic staff of the School of Computing and Informatics (in particular Dr. Richard Cant) who have answered my many questions and the Tech Support guys (especially Mark Howson) for their assistance as required over the last 4 years.

Thanks also to my office buddies Dr. Adam Watkins, Dan Rhodes and Jon Townsend for making the days pass far too quickly and the rest of the post-graduate students to whom I probably owe many cups of coffee. (I would also like to thank the coffee producers of Costa Rica for keeping me thinking and the Fair Trade Foundation for making me feel okay about drinking so much coffee).

Still yet more thanks to my friends, wherever you are, thanks for being there, but especially Steph, Rob and my brother Tim for helping me get through some really difficult times and for all the emotional support, camaraderie, entertainment and for showing they care.

Lastly, and most importantly, I have to say thank-you to my family, particularly my Mum and Dad without their love and support this would never have been completed (or possibly even started). To them I dedicate this thesis. Thanks folks.



# **Table of Contents**

<b>Abstract</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>Table of Contents</b>	<b>iv</b>
<b>Table of Figures</b>	<b>ix</b>
<b>Chapter 1 - Introduction</b>	<b>1</b>
<b>1.1 Data Acquisition for Reverse Engineering</b>	<b>3</b>
1.1.1 Distortion Problems in the Data Acquisition Process	3
<b>1.2 Previous Work</b>	<b>5</b>
1.2.1 Compensation Method Based on Multiple Scans	5
1.2.2 Comparison of Matsushita ANR-1182 with Wolf & Beck Laser	7
<b>1.3 Project Aims</b>	<b>10</b>
1.3.1 Objectives	11
1.3.2 Contributions to Knowledge	12
1.3.3 Definition of an Edge	13
<b>1.4 Organisation of Thesis</b>	<b>14</b>
<b>Chapter 2 - Reverse Engineering</b>	<b>16</b>
<b>2.1 Other Applications for the 3D Digitisation Process</b>	<b>17</b>
<b>2.2 The Reverse Engineering Process</b>	<b>19</b>
<b>2.3 Data Acquisition Methods</b>	<b>20</b>
2.3.1 Tactile Methods	22
2.3.2 Non Tactile Methods	23
2.3.3 Optical Data Acquisition	24
2.3.4 Active Optical Methods	24
<b>2.4 Common Problems in Data Acquisition</b>	<b>33</b>
2.4.1 Limitations of Single Point Laser Triangulation Sensors	34
2.4.2 Causes of Occlusion and Secondary Reflection Errors	39

<b>2.5</b>	<b>Summary</b>	<b>51</b>
<b>Chapter 3 - Image Processing</b>		<b>53</b>
<b>3.1</b>	<b>Image Kernels and Convolution</b>	<b>54</b>
<b>3.2</b>	<b>Image Smoothing</b>	<b>54</b>
<b>3.3</b>	<b>Edge Detection</b>	<b>55</b>
3.3.1	Definition of an Image Edge	55
3.3.2	First Derivative (Gradient) Edge Detectors	56
3.3.3	The Canny Edge Detector	57
3.3.4	Second Derivative Detectors	59
3.3.5	Assessment of Edge Detection Algorithms	59
3.3.6	Corner Feature Detection	60
<b>3.4</b>	<b>Thresholding</b>	<b>61</b>
3.4.1	Thresholding as a Means of Labeling Edges	61
<b>3.5</b>	<b>Region Detection &amp; Texture Analysis</b>	<b>63</b>
<b>3.6</b>	<b>Image Analysis</b>	<b>64</b>
3.6.1	Vectorisation	64
3.6.2	Hough Transform	64
<b>3.7</b>	<b>Summary</b>	<b>67</b>
<b>Chapter 4 - Improved Scanning Using Data Fusion Based On Image Analysis</b>		<b>68</b>
<b>4.1</b>	<b>Development of Conceptual ‘Data Fusion’ Solution</b>	<b>69</b>
4.1.1	The Concept of ‘Orientation Alignment Tolerance’	70
4.1.2	Method 1 - Dominant Scan Orientation	72
4.1.3	Method 2 Algorithm – Multiple Partial Scans	73
4.1.4	Method 3 Single Scan with Rotating Scan Head	79

<b>4.2</b>	<b>Hardware Requirements</b>	<b>80</b>
4.2.1	Current Hardware	80
4.2.2	New Hardware Requirements	86
<b>4.3</b>	<b>Calibration &amp; Registration Issues</b>	<b>92</b>
4.3.1	Laser Calibration	92
4.3.2	Image Registration and Scaling	104
4.3.3	Simple Corrections for Camera Scale, Skew and Rotation	111
<b>4.4</b>	<b>Other Systemic Distortions</b>	<b>116</b>
4.4.1	Synopsis of Hardware Operation	116
4.4.2	Timing Signal Noise	117
4.4.3	Operating System Latency	118
4.4.4	Noise on the Laser Analogue Data Line	122
<b>4.5</b>	<b>Summary</b>	<b>126</b>
<b>Chapter 5 - Edge Detection and Vectorisation Methods for Improved Scanning</b>		<b>128</b>
<b>5.1</b>	<b>Image Processing Algorithms</b>	<b>129</b>
5.1.1	Greyscale vs. Colour Images and Image Resolution	129
5.1.2	Software Framework	131
5.1.3	Software Overview	132
5.1.4	Edge Detection Algorithms	133
5.1.5	Evaluation of Image Processing Algorithms	138
<b>5.2</b>	<b>Vectorisation Algorithms</b>	<b>151</b>
5.2.1	Hough Transform	151
5.2.2	Vectorisation Using Least Squares Algorithm	154
<b>5.3</b>	<b>Summary</b>	<b>158</b>

<b>Chapter 6 - Data Fusion Algorithms</b>	<b>160</b>
<b>6.1 Software Framework</b>	<b>161</b>
6.1.1 The Scan Acquisition Process	164
6.1.2 Image Scan Calibration	166
<b>6.2 Experimental Scan Region Generation Algorithms</b>	<b>167</b>
6.2.1 Dominant Single Orientation Scan	167
6.2.2 Partial Scan by Region Orientation	169
6.2.3 Scan Orientation By Nearest Vector Orientation	176
<b>6.3 Scan Data Interpretation</b>	<b>179</b>
6.3.1 Loading - Data Regularisation / Interpolation	180
6.3.2 Display / Operator Interaction	181
6.3.3 Scan Time Estimation	182
<b>6.4 Summary</b>	<b>183</b>
<b>Chapter 7 - Experimental Results</b>	<b>184</b>
<b>7.1 Methods for Evaluation of Results</b>	<b>187</b>
<b>7.2 Example of the Scan Acquisition Process</b>	<b>188</b>
7.2.1 Domino Calibration	189
7.2.2 Domino Edge Detection Stage	190
7.2.3 Domino - Vectorisation Stage	191
7.2.4 Domino – Scan Selection by Scan Region Orientation	194
7.2.5 Domino – Scan Selection by Nearest Vector Orientation	195
<b>7.3 Surface Reflectivity Issues</b>	<b>196</b>
<b>7.4 Distortions Due to Camera Alignment and Optics</b>	<b>199</b>
7.4.1 Gauge Plate	200
7.4.2 Summary of Image Distortion Problems	207
<b>7.5 Overlapping Regions</b>	<b>208</b>

<b>7.6</b>	<b>Edge Curves and Corners</b>	<b>212</b>
7.6.1	Low Curvature (Bottle Cap)	212
7.6.2	High Curvature (Triangle Block)	219
7.6.3	Corners	226
<b>7.7</b>	<b>‘Real World’ Objects</b>	<b>227</b>
7.7.1	Giraffe Cookie Cutter	227
7.7.2	Bronte Plaque	234
<b>7-8</b>	<b>Summary of Results</b>	<b>246</b>
<b>Chapter 8 - Future Work</b>		<b>248</b>
<b>8.1</b>	<b>Image Analysis</b>	<b>248</b>
<b>8.2</b>	<b>Improved Vectorisation Methods</b>	<b>249</b>
<b>8.3</b>	<b>Compensation for ‘Irreconcilable’ Scan Data Points</b>	<b>250</b>
<b>8.4</b>	<b>Other Methods</b>	<b>252</b>
8.4.1	Pre-Scan Profiling	252
8.4.2	Material Reflectivity Index	253
8.4.3	‘Just in Time’ Edge Detection	254
8.4.4	Mapping of image data to 3D Data Object	254
8.4.5	Further User Interaction	255
8.4.6	Artificial Intelligence Approaches	255
<b>Chapter 9 – Conclusions and Project Summary</b>		<b>256</b>
<b>Table of References</b>		<b>262</b>
<b>Appendix A – Derivation of Calibration Algorithm</b>		<b>A-1</b>
<b>Appendix B – Submitted Book Chapter</b>		<b>B-1</b>
<b>Appendix C – Prep’04 Conference Paper</b>		<b>C-1</b>
<b>Appendix D – ICCMS 2005 Conference Paper</b>		<b>D-1</b>

## Table of Figures

Figure 1-1	Overview Of The Reverse Engineering	2
Figure 1-2	Effects Of Changing Sensor Orientation	6
Figure 1-3	Comparison Of Wolf & Beck And Matsushita Scanners (Single Line Cross-Section)	8
Figure 1-4	Comparison Of Wolf & Beck And Matsushita Scanners (Giraffe Cookie Cutter Mould)	9
Figure 2-1	Overview Of The Main Steps In The RE Process.	19
Figure 2-2	Taxonomy Of Digital Shape Acquisition Methods.	21
Figure 2-3	Tactile Sensor Methods	22
Figure 2-4	A Taxonomy Of Optical Data Acquisition Techniques	24
Figure 2-5	Pulse ToF Laser System	26
Figure 2-6	Moiré Interferometry Pattern	27
Figure 2-7	Changes In Triangulation Angle With Changes In Distance.	28
Figure 2-8	Internal Functions Of A Single-Perspective Point Sensor.	31
Figure 2-9	Single Point Laser Scanner Employing Longitudinal Synchronisation	31
Figure 2-10	Line (Or ‘Slit’) Laser Scanner.	32
Figure 2-11	Comparison Of Specular And Diffuse Surface Reflectance	35
Figure 2-12	Beam Diameter vs. Measurable Feature Size	36
Figure 2-13	Laser Beam ‘Waist’	37
Figure 2-14	Bow-Wave Distortions Increase With Edge Gradient	40
Figure 2-15	The Nature Of Occlusion Spikes Changes With Slope Gradient	41
Figure 2-16	Cause Of Occlusion Errors	43
Figure 2-17	Cause Of Secondary Reflection Errors	45
Figure 2-18	Variation In Distortion For Different Sensor Orientations And Edge Heights	49
Figure 2-19	Shape Capture With Sensor Perpendicular To Object	50

Figure 3-1	Idealised Edge Response Types	56
Figure 3-2	Canny Edge Detection On Tile	58
Figure 3-3	Hough Parametric Description of Straight Line	65
Figure 3-4	Hough Parametric Space 2D Accumulator Array	65
Figure 3-5	Hough Transform of Canny Edge Detected Lines	66
Figure 4-1	Laser Orientation Tolerance	71
Figure 4-2	Scan Region Orientations Applied To A Simple 'L' Shaped Object	75
Figure 4-3	Partial Scan Regions By Nearest Vector	78
Figure 4-4	Image Showing The Major Components Of The Test Rig	81
Figure 4-5	Diagrammatic Representation Of The Laser Sensor, Showing Main Components.	83
Figure 4-6	Diagram Showing Voltage Fluctuation Representing Range Of Laser	84
Figure 4-7	Matsushita NAIS ANR1182 Sensor In Use, Scanning An Inscribed Metal Block.	85
Figure 4-8	Diagram Showing Views Of Laser Sensor Mounting On Stepper Motor Axle	87
Figure 4-9	Schematic Of CNC Machine Showing Location Of Lights And Reflector Boards.	89
Figure 4-10	Correct Laser Alignment.	93
Figure 4-11	Laser Rotation With Offset	94
Figure 4-12	Laser Tilted Relative To The Plane Of The Scan Bed.	96
Figure 4-13	Laser Sensor Tilt Affects The Measured Height And Position	97
Figure 4-14	Sensor Tilt With Emitter Offset From Axis Of Sensor Rotation.	99
Figure 4-15	Generalised Scenario With Axis Of Rotation And Laser Sensor Both Oriented At Arbitrary Angle To Scan Bed	101
Figure 4-16	View Of The Tilted Scanner Aligned With The Edge Of An Object	103
Figure 4-17	Camera Perspective Distortions	106
Figure 4-18	Camera Parallax Error	107
Figure 4-19	Camera Pincushion And Barrel Distortion	108

Figure 4-20	Lens Barrel Distortion Of Webcam Image	108
Figure 4-21	Comparison Of Distortions In Webcam And Digital Camera	110
Figure 4-22	Comparison Of Webcam And Digital Camera Distortions: Compression By 95% In Y Axis	111
Figure 4-23	Changes In Camera Position Affect The Object's Image.	112
Figure 4-24	Calibration Markers Placed Around Scan Object.	114
Figure 4-25	Calibration Dialog Window Showing Corresponding Image And Scan Coordinate Pairs	115
Figure 4-26a)	Noise Spike On Timing Signal Line Causes Early Data Collection Trigger	119
Figure 4-26b)	Latency In Task Switching Causes Delay In Digital Input Timing Signal	120
Figure 4-26c)	Corrected Timing Diagram: Addition Of Axcontroller Wait Timer	121
Figure 4-27	Noise On Laser Data Signal.	123
Figure 4-28	Typical Variation In Values Caused By Signal Noise On The Sensor Data	125
Figure 5-1	Overview Of Proposed Software Design For Image Processing Stage	132
Figure 5-2	Operator Interaction Required In Selecting Edge Detector Parameters	134
Figure 5-3	Table Showing Comparative Time To Execute First Derivative Operators	139
Figure 5-4	Comparing The Output Of The First-Derivative Gradient Operators.	140
Figure 5-5	Mean Execution Time For First Derivative Compass Operators	142
Figure 5-6	Comparing The Output From The Prewitt And Kirsch Compass Operators With The Standard Prewitt Kernel	143
Figure 5-7	Results From Canny Edge Detector	146
Figure 5-8	Results From Second Derivative Operators	149
Figure 5-9	Windowed Hough Transform	153
Figure 5-10	Recursive Windowed Hough Transform	153



Figure 6-1	Overview Of Software Design Data Flow	161
Figure 6-2	Overview Of Software Design As Implemented For Testing	163
Figure 6-3	Corresponding Images For Calibration And Analysis.	166
Figure 6-4	Concept Of The Orientation Image ‘Stack’	171
Figure 6-5	Scan Region Width Calculation	173
Figure 6-6	Gcode File Set Import Dialog	181
Figure 7-1	Comparison of Processing & Scan Times for Test Objects	186
Figure 7-2	Domino	188
Figure 7-3	Domino Calibration.	189
Figure 7-4	Domino Canny Edge Detection (High Threshold)	190
Figure 7-5	Domino Canny Edge Detection (Lower Threshold)	191
Figure 7-6	Domino Vectorisation (Vectors Only)	192
Figure 7-7	Domino Vectorisation (Vector Overlay)	193
Figure 7-8	Domino Scan Region Map (SRO Method)	194
Figure 7-9	Domino Scan Region Map (NVO Method)	195
Figure 7-10	Unsprayed Domino Results	197
Figure 7-11	Sprayed Domino Results	198
Figure 7-12	Domino Results (NVO Method)	199
Figure 7-13	Gauge Plate Scan Region Maps	200
Figure 7-14	Gauge Plate Scan Results (Webcam, NVO Method)	202
Figure 7-15	Gauge Plate Scan Results (Digital Camera, NVO Method)	202
Figure 7-16	Gauge Plate NVO Scan Region Map Manual Alignment	203
Figure 7-17	Manually Corrected Gauge Plate Scan Results (Webcam, NVO Method)	204
Figure 7-18	Manually Corrected Gauge Plate Scan Results (Digital Camera, NVO Method)	205
Figure 7-19	Table of Root Mean Square Errors for Different Scan Methods on Gauge Block	206

Figure 7-20	Gauge Plate Scan Region Orientation Region Map Showing Overlaps	208
Figure 7-21	Gauge Plate Scan (SRO, Digital Camera) Results Showing Overlap Errors	209
Figure 7-22	Plan View Of Gauge Plate SRO Combined Scan With Overlaps Marked	210
Figure 7-23	Bottle Cap Image	213
Figure 7-24	Bottle Cap SRO Scan Region Map	214
Figure 7-25	Bottle Cap ‘Best’ Single Scan Orientation	215
Figure 7-26	Bottle Cap Results Of SRO Method	216
Figure 7-27	Bottle Cap NVO Scan Region Map	217
Figure 7-28	Bottle Cap Results Of NVO Method	218
Figure 7-29	Triangle Block Image	220
Figure 7-30	Triangle Block Scan Region Maps	220
Figure 7-31	Triangle Block ‘Best’ Single Orientation	222
Figure 7-32	Triangle Block Results From SRO Method	222
Figure 7-33	Triangle Block Results From NVO Method	223
Figure 7-34	Triangle Block Comparison Of SRO And NVO Methods	224
Figure 7-35	Triangle Block Scan Cross Section	225
Figure 7-36	Giraffe Cookie Cutter Image Edge Detection	227
Figure 7-37	Giraffe Cookie Cutter Vectorisation	228
Figure 7-38	Giraffe Cookie Cutter NVO Scan Region Map	229
Figure 7-39	Giraffe Cookie Cutter Single Orientation Scan (Side View)	231
Figure 7-40	Giraffe Cookie Cutter Single Orientation Scan (Oblique View)	231
Figure 7-41	Giraffe Cookie Cutter NVO Combined Scan Result (Side View)	232
Figure 7-42	Giraffe Cookie Cutter NVO Combined Scan Result (Oblique View)	232
Figure 7-43	Brontë Plaque Image	235
Figure 7-44	Brontë Plaque Edge Detection	236
Figure 7-45	Brontë Plaque Vectorisation	238

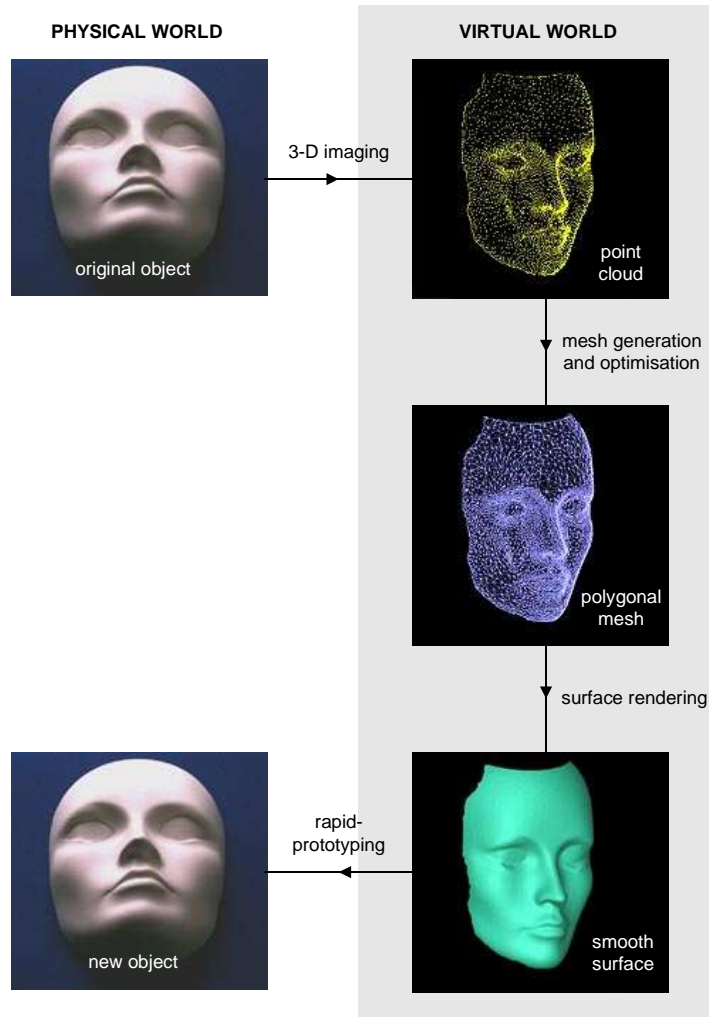
Figure 7-46	Brontë Plaque NVO Scan Region Map	239
Figure 7-47	Brontë Plaque Single Best Orientation Scan (Top View)	240
Figure 7-48	Brontë Plaque Single Best Orientation Scan (Side View)	241
Figure 7-49	Brontë Plaque NVO Combined Scan (Top View)	242
Figure 7-50	Brontë Plaque NVO Combined Scan (Side View)	242
Figure 7-51	Brontë Plaque NVO Combined Scan (End View)	243
Figure 7-52	Brontë Plaque NVO Combined Scan (Oblique View)	243
Figure 7-53	Brontë Plaque NVO Combined Scan Cross Section (Front)	244
Figure 7-54	Brontë Plaque NVO Combined Scan Cross Section (Back)	244
Figure 7-55	Brontë Plaque Single Line Scan Cross Section	245
Figure 8-1a)	Internal Corner Errors	251
Figure 8-1b)	Internal Corner Error Compensation	252

# **1. Introduction**

Engineering encompasses the design, manufacture, construction, and maintenance of products, systems and structures. In the traditional engineering process, a design is first conceptualised and then modified, using either technical drawings on paper or a digital representation on a Computer Aided Design (CAD) system before production of the physical system or a tangible prototype. For example, in manufacturing a new part, the designer would first sketch an idea on a piece of paper. In the past this design would be formalised by a draughtsman on paper with accurate dimensioning. Modern designers have computers and software such as AutoCAD to illustrate the design more clearly. This design would then be realised by either handcrafting the part (e.g. use of a manually operated lathe by a skilled operator) or feeding its digital representation into a highly precise cutting machine, such as a computer numerical control (CNC) machine. The finished physical product should closely resemble the original design.

The process described above is the traditional ‘forward engineering’ approach to producing an object and has been common practice for hundreds of years. There is an alternative approach: that of reverse engineering, which begins with a physical representation of the object that the designer wishes to recreate as a design, either in order to replicate the original object (e.g. if a design has been lost or is otherwise unavailable) or as a basis for a modified design. Reverse engineering involves the analysis of an object or system to examine how it functions in order to duplicate or enhance the object or system. Figure 1-1 shows an overview of the reverse engineering process.

Reverse engineering may be seen as the process of analysing an existing system to identify the system’s components and their inter-relationships, to establish forms of representation of the system at a higher level of abstraction and to produce or replicate that system. The application of reverse engineering techniques is discussed in more detail in Chapter 2 of this thesis.



**Figure1-1: Overview of the Reverse Engineering Process [1]**

Within the scope of the work described in this thesis the primary concern is with the part of this process referred to in Figure 1-1 as 3D Imaging, or the data acquisition phase. The choice of an appropriate method of data acquisition can have a great effect on the quality of the final model. It is therefore of importance to investigate how to minimise errors within this phase of the reverse engineering process.

## **1.1 Data Acquisition for Reverse Engineering**

Numerous methods exist for the acquisition of shape information. These methods can be divided broadly into ‘tactile’ methods, which involve the use of a contact sensor that is positioned on the surface of the sample object either by human or machine operation, and ‘non-tactile’ methods which involve the measurement of some form of energy (e.g. light or sound waves) falling onto the object under consideration. Tactile methods are accurate but have the disadvantage of being very slow. Non-tactile methods are generally faster but subject to a number of issues often related to the type of energy used by the system. Non-tactile methods may be further divided into ‘active’ and ‘passive’ systems depending on whether the energy source used is projected by the measuring device or whether it makes use of ambient energy.

Commonly used methods involve the use of active range-finding devices, which measure the position of each point on the object’s surface relative to a known position. Many rangefinders use optical methods and measure either the time-of-flight or the triangulation angle between the emitter and detector. Such rangefinders can acquire regular, dense sample sets known as range images. The resulting data is represented as a point cloud<sup>i</sup>. Further processing is required in order to generate efficient and reliable computer models from these sample points.

### **1.1.1 Distortion Problems in the Data Acquisition Process**

In order to digitise object shapes for this project the popular range scanning method known as laser triangulation has been used. A laser beam is projected onto the surface of the object and the angle at which the light returns from the object allows the distance to be calculated.

Single perspective laser sensors provide a relatively cheap and simple method of data acquisition. However fundamental limitations in the accuracy of this method are caused by effects such as the occlusion and secondary reflection of the laser beam by the geometry of the object, resulting in distortions in the data.

---

<sup>i</sup> A point cloud is a collection of 3-D coordinates in a Euclidean coordinate space. Each point represents one or more data samples. (Multiple samples are averaged by the sensor to provide the data for each point, with the number dependent on the speed of the scanning process).

The distortions from the laser scanner may be categorised as either stochastic or systematic. Stochastic distortions, also known as noise, may occur because of random inaccuracies in measurement, such as mechanical vibration of the platform or a weak source of illumination. Noise reduction is possible through the application of post-processing techniques to the range images. Methods such as wavelet transform that allows multi-resolution feature selection and extraction [2] or smoothing based on an approximation of a mesh by circular arcs [3] can be employed. However these methods often result in some degree of smoothing in the final data.

Systematic distortions occur if there is an error in the calibration of the system. For example, selection of incorrect scale or offsets will lead to consistently false distance measurements. Surface properties of the object, such as highly contrasting surfaces, may also cause systematic distortions. These can be prevented by ensuring homogeneity in surface reflectivity through treating the object with a non-permanent coating prior to any imaging.

A common cause of systematic distortion with triangulation sensors is related to the geometry of the object. The occlusion of the light beam by another part of the object can produce significant errors within certain regions of the scan. Smaller errors can be caused by secondary reflections of the light beam from other parts of the object. Wong [1] found that these geometric errors can be minimised by changing the orientation of the laser sensor relative to geometric edges within the object. This approach can also minimise errors caused by transitional edges<sup>ii</sup>.

Strategies to reduce systematic distortions due to sensor's orientation include planning the optimal orientation and path for laser scanning [4] or employing sensors with multiple detectors with intrinsic averaging [5]. Another procedure for removing significant distortions is to simply trim them and replace the discarded values with new interpolated or averaged values. Such an approach, however, will also inevitably remove any detail in those trimmed regions.

---

<sup>ii</sup> Transitional edges are those caused by an abrupt change in colour or surface reflectivity not associated with a change in the object's geometric profile.

## **1.2 Previous Work**

Wong[1] explored interactive and automated approaches to reduction of data acquisition distortions. He investigated how to reduce the errors caused by the object geometry such as occlusion of the laser beam by another part of the object. In order to obtain a good digital representation of the object of interest, the laser scanner needs to be oriented parallel to the geometric edges of the object. This may be done by the operator manually selecting different scan orientations for different parts of the object. In order to avoid the requirement for human interaction Wong developed a method by which the errors could be recognised (by comparing scans at a number of different orientations) and used to determine the edge position and orientation automatically, thereby enabling selection of the data from the closest orientation.

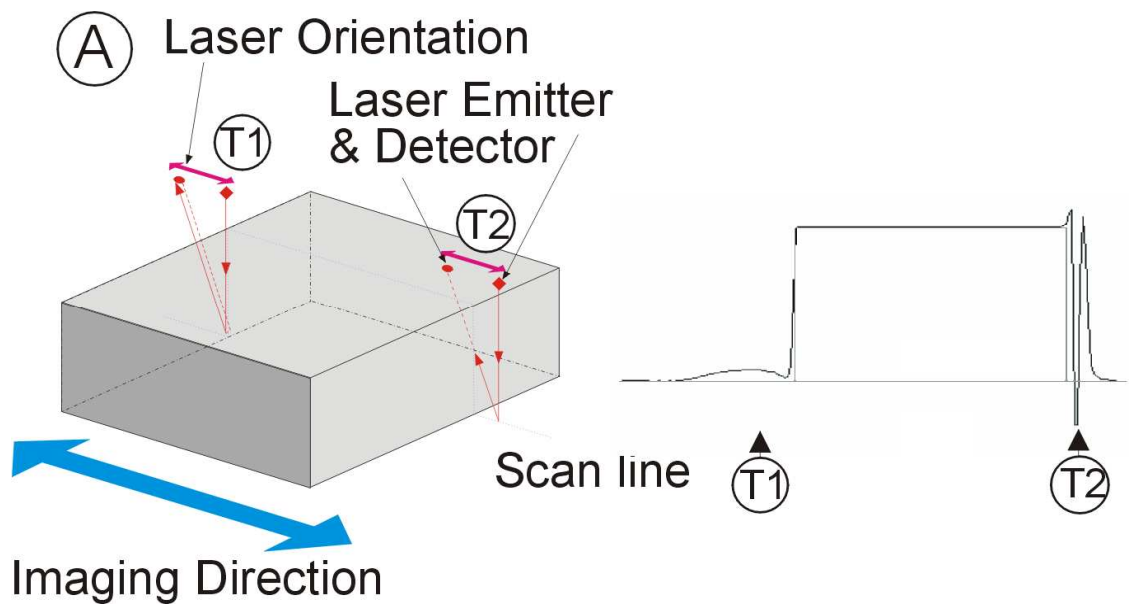
### **1.2.1 Compensation Method Based on Multiple Scans**

Wong[1] demonstrated that the orientation of the laser scan head (rather than the scan direction) was critical in the degree of distortions during the scanning process. Errors caused by the object's geometry are at their worst when the scan head is oriented perpendicular to the object being scanned, as shown in Figure 1-2(a).

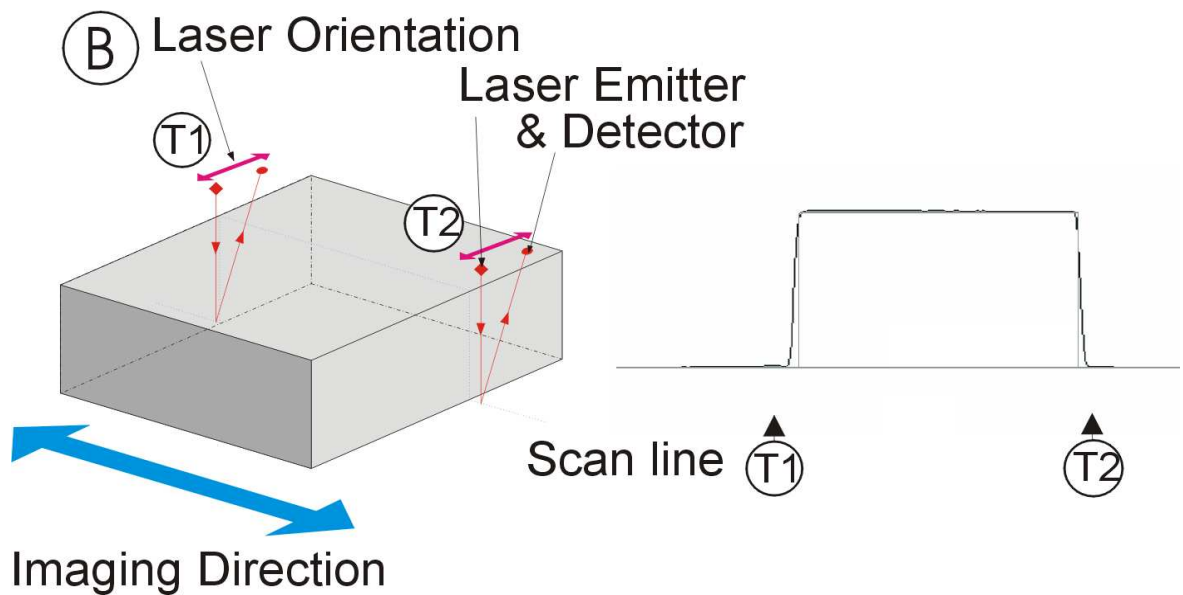
Secondary reflections at T1 cause a 'bow wave' effect, while at T2 'spikes' are caused by occlusion of the laser beam. Conversely, if the laser scan head is oriented parallel to the 'edges' of the object then the errors can be minimised, as shown in Figure 1-2(b). At both T1 and T2 the detected position of the laser spot is unaffected by other parts of the object's geometry.

In order to address these errors, Wong's approach was to use a number of complete scans of the object with the scan head at different orientations. Software was developed to identify error regions and determine the corresponding edges. Based on the edge orientation the software selects the part of each scan where the error level is minimal and combines the resulting partial range images. Wong's method requires at least 8 complete scans of the object to be performed before his processing method can begin. Unfortunately the software Wong developed and the data he gathered is no longer available for direct comparison.





**Figure 1-2(a):** Illustration of scanning errors when the laser is oriented perpendicular to the edge of the block. At T1 secondary reflections from the vertical edge cause the average position of the detected spot to be shifted on the sensor ('bow wave'). At T2 the detector is occluded by the geometry of the object causing an 'out-of-range' response. Spikes result from partial occlusion and secondary reflections.



**Figure 1-2(b):** When the laser is oriented parallel to the edges of the object distortions are minimised across the object scan on both sides of the object.

Whilst this method results in much improved scan quality, the cost of this approach is a greatly increased overall scan time, increased data storage requirements and considerable processing costs. For example, a scan area of 10cm x 10cm requires approximately 3½ hours to scan using the Matsushita laser at a coarse (0.5mm) scan resolution (approximately 25 minutes per scan, not including the overhead of registering between the scans). In many cases a much higher scan resolution is required in order to generate a more detailed surface model (e.g. a scan-point resolution of 0.05mm would require 10 times as many scan lines to be recorded). Higher resolutions require significantly more scan lines and a slower scanning speed. Therefore the time for an individual scan is increased, resulting in total scan times that are unacceptably long for the ‘complete’ set of scans required by Wong’s method.

A method is therefore required by which the locations where distortions are likely to occur for a particular orientation may be identified *prior to scanning*. Then the occurrence of distortions can be avoided, instead of compensating for them after the data has been collected. Although the acquisition time for the 8 required scans proved commercially unacceptable the basic idea of determining edge orientation and selecting the object height data based on the appropriate scan orientation is the basis of the work described in this thesis.

### **1.2.2 Comparison of Matsushita ANR-1182 with Wolf & Beck Laser**

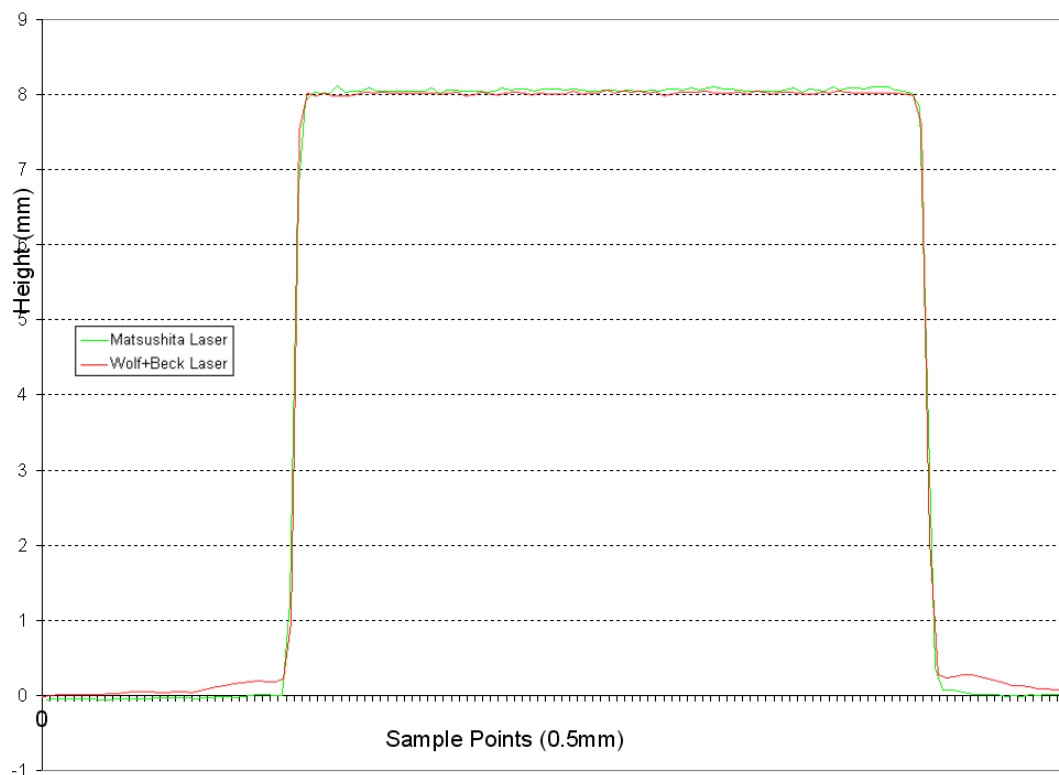
Wong [1] had access to two different laser scanners: the Matsushita ANR1182 scanner, and the Wolf & Beck OTM3A-50 scanner. Both units employ active triangulation however the Matsushita scanner has a single emitter and single photo-sensitive detector, whereas the Wolf & Beck scanner employs multiple sensors spaced equally in a circle around the emitter and employs internal (hardware) averaging between the sensors to give a single output value.

This arrangement of sensors in the OTM3A-50 is usually successful in removing occlusion data spikes. Whereas the Matsushita scanner would report ‘out of range’ and generate a ‘spike’ response, the Wolf & Beck laser avoids this problem because a return signal is received at some of the detector elements (unless the emitter is in an ‘internal hole’ situation). The calculated data point may not be absolutely identical to the true surface position but it is a lot better than the ‘out of range’ response and data spikes generated by the ANR-1182.

However the added complexity of the Wolf & Beck is reflected in the unit cost of the scanner. At current (2006) prices the unit cost of the Wolf & Beck OTM3A-50 scanner is approximately £5000 compared to approximately £1000 for the Matsushita ANR1182.

Furthermore, the averaged output from the OTM3A-50 is not *always* as ‘correct’ as the ANR1182. Although the multiple detector arrangement of the W&B scanner means that some return signal is almost always detected, this arrangement can cause a ‘bow wave’ response to be generated in certain situations due to the intrinsic averaging process in the laser unit (see Figure 1-3). Because the detectors are evenly distributed around the emitter, rotating the laser makes no difference to the quality of the output.

In the same situation the Matsushita laser, having only a single detector, would not generate a bow wave if the unit was parallel to the edge. That is to say, at its ‘best’ orientation the single-detector triangulation sensor is as good as, or better than one that employs multiple detectors; the problem lies with correctly aligning the sensor with the geometric edge.



**Figure 1-3: Comparison of Matsushita and Wolf & Beck Scanners - Cross section of Gauge Block**  
The Masushita sensor is oriented parallel to the edge and displays no significant distortion. The Wolf & Beck sensor shows a small ‘bow wave’ distortion on both sides of the block.

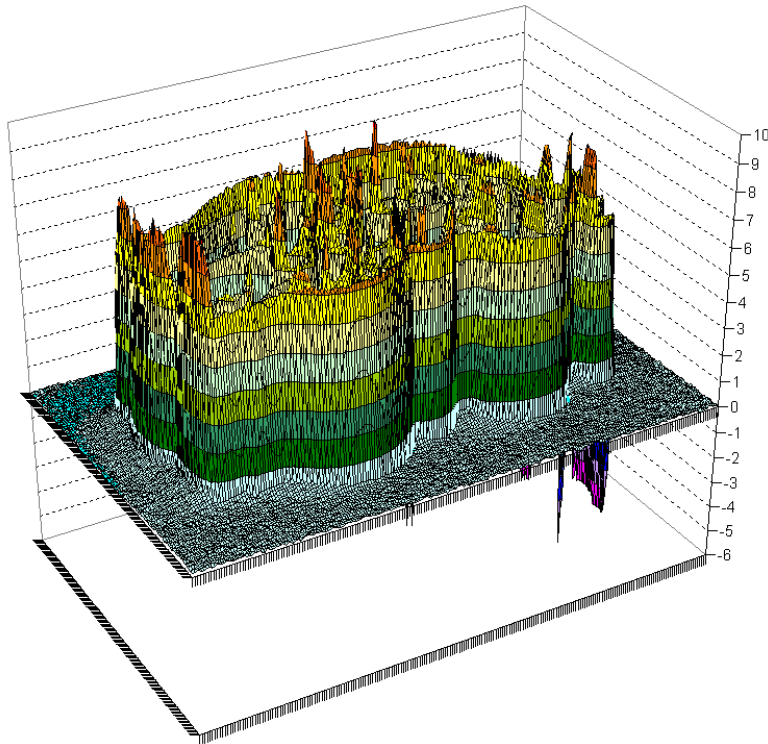
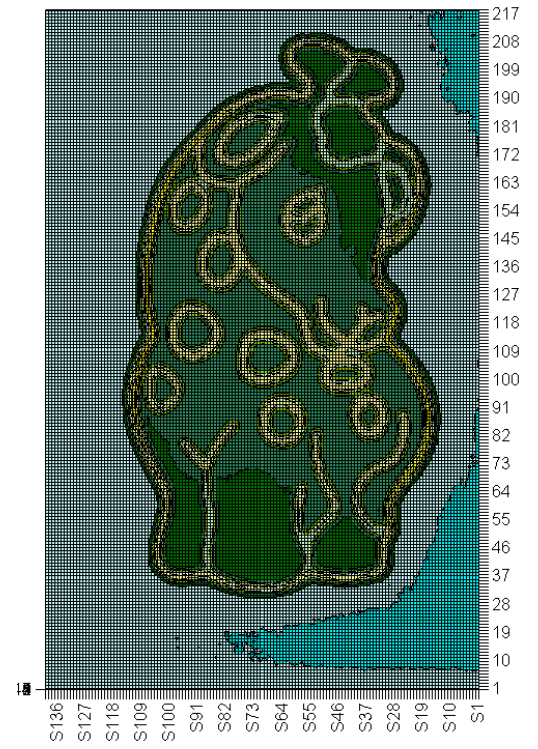
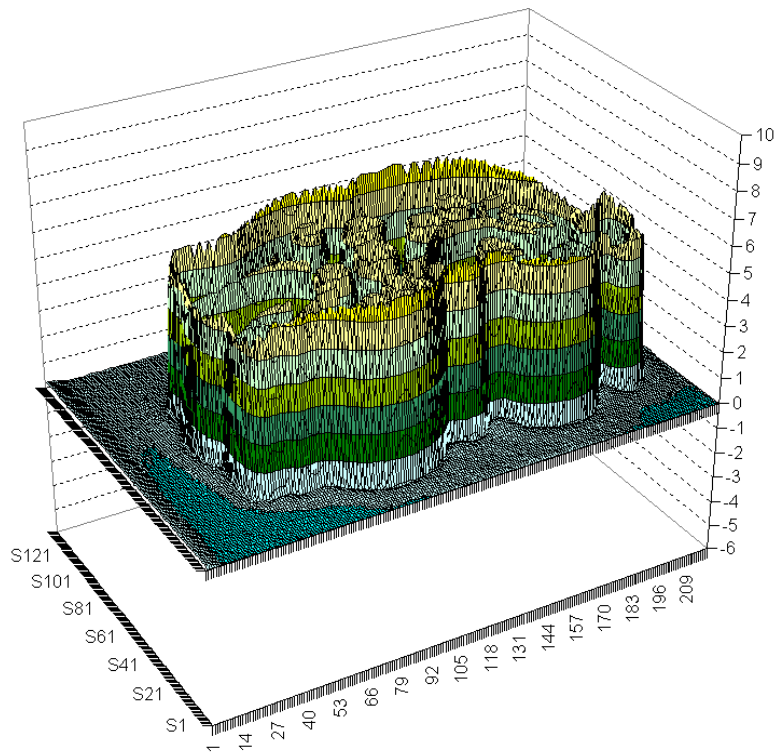


Figure 1-4: Comparison of Wolf & Beck (top) and Matsushita (bottom) sensors on the Giraffe Cookie Cutter Mould. The multiple-perspective detectors of the Wolf&Beck unit avoid the occlusion errors (data spikes) to which the Masushita sensor is vulnerable.

### **1.3 Project Aims**

The aim of the work described in this thesis is to investigate how distortions in range images can be minimized by integrating an optical camera with the laser scanner in order to generate an optimal scan path. Sufficient *a priori* knowledge of the geometry, location, and orientation (relative to the scanner) of the object under consideration will allow the quality of the scan acquired by a range scanner to be improved whilst minimising the time necessary to complete the scan.

The information acquired from photographs and range images is complementary. Range images can capture intricate surface detail, whereas an optical image provides an ‘overview’ of the object. The basic concept of camera technology has not changed significantly in the last century, although the advent of digital imaging technology has effectively replaced the chemical-based film with an electronic CCD imaging plane that can store the captured image electronically. Developments in digital imaging technology over recent years have seen a rapid rise in cheap, commercially available devices.

To achieve the aim stated above, it is necessary to:

- Integrate information acquired from the optical camera with the laser scanner.
- Use image analysis methods to find regions where distortions are likely to occur in the laser scan by identifying edges in a corresponding photographic image of the object.
- By recognising problem areas prior to the laser scan process, determine an optimised scanning path with respect to sensor orientation, thereby improving the quality of the acquired scan data whilst minimising the increase in required scan time.

One of the key aspects of this project is the overall cost of the system components (which should not exceed that of more expensive laser scanning systems such as the Wolf & Beck scanner). Thus the investigation has centred on what may be achieved using cheap and easily accessible imaging devices such as webcams and ‘domestic’ digital cameras.

### 1.3.1 Objectives

The proposed system will determine the optimal scanning orientation (i.e. with the laser oriented parallel to edges) prior to any scans being performed. The objectives of this project provide a series of logical steps in order to achieve the aims as stated above. These tasks may be summarised as follows:

- Establish a development platform to investigate how image analysis techniques can be used to plan the scanning process.
- Develop an understanding of the occurrence and causes of distortions in laser triangulation scanning.
- Investigate current ‘state of the art’ in image analysis techniques and develop an evaluation platform.
- Establish a system that integrates optical camera information with the laser scan data, including methods of calibration.
- Develop image analysis algorithms, employing a selection of edge and region detection methods based on current popular methods, such as the Canny edge detector.
- Evaluate the effectiveness of the edge detection algorithms for the type of images generated by the system.
- Develop algorithms which use the output from the image analysis techniques to determine the ‘locally optimal’ scan orientation in order to minimise distortions in the scan.
- Develop algorithms to merge the results from the chosen image analysis methods into a single, integrated scan path plan with locally optimised sensor orientation.
- Evaluate the improved scanning path algorithms by comparing their accuracy against scans of a single orientation.

An optical camera system is used to capture an image of the object and then edge detection methods are employed, followed by vectorisation in order to determine the position of significant edge vectors.

These vectors are then used to develop a scan path over the object. The idea is to use the edge as an ‘exemplar’ at a given position and to develop a scan region around the edge, with the same orientation as the edge. There may also be many areas in the image where the orientation of the scan head is non-critical if there are no edges nearby. ‘Scan orientation regions’, corresponding to different scanner orientations, can thus be generated based on these detected edges in order to minimise the potential errors due to occlusions and secondary reflection.

One strategy is to perform a single complete scan of the object and then rescan critical regions at other orientations. It is likely that, for most objects, the ‘scan regions’ will have some partial overlap, therefore the total scan time will be somewhat greater than for a ‘single pass’ scan but still much shorter than Wong’s 8 complete scans. The advantage of much improved overall quality will therefore be achieved without the overhead of the protracted scan time.

### **1.3.2 Contributions to Knowledge**

1. Use of (single) optical camera image to provide information for sensor orientation.
2. Development of ‘scan orientation region’ concept based on proximity to vector.
3. Development of ‘generic convolution template’: capable of accepting any defined convolution matrix and applying the same sequence of operations thus providing a common code base reference to compare different operators.
4. Development of a new least-squares method of line-growing vectorisation.

### 1.3.3 Definition of an Edge

In this thesis the word ‘edge’ is used to describe features in both 2D and 3D images. An edge in the optical (camera) image appears as a discontinuity (i.e. a sudden change) in colour or intensity. Canny & Deriche define an edge as the maxima of the gradient modulus in the gradient direction. In a 3D model an edge is the intersection of 2 surfaces, however this applies only to ‘sharp’ edges – in the real world surfaces often ‘blend’ or merge together and it is difficult to pinpoint the precise position of the edge – in this case the edge may be defined as a position of high curvature of the surface or more precisely as a position of maximum change in direction of the tangent to the surface. A region that is simply of an arbitrary constant curvature (e.g. a cylinder) may also appear as an ‘edge’ dependent on the viewing angle.

Geometric (3D) edges will usually show a response in a 2D image, although there may be lighting or contrast difficulties that cause the position of the edge to be lost. Controlled lighting can help to rectify this problem. Some 2D edges that have no 3D profile will also be detected by the optical image camera. Such edges may be boundaries between regions of differing reflectivity. Some edges in the photographic image are ‘false’ edges caused by shadows cast by the object. Shadows are considered as a hindrance in detecting ‘true’ geometric edges (i.e. they give ‘false positive’ edge responses) and will be avoided by the use of controlled lighting wherever possible. The use of shadow removal algorithms may be considered.



## **1.4 Organisation of Thesis**

This thesis is structured into nine chapters. In this Chapter an overview of the concepts central to the work in this thesis is provided and the aims and objectives of this project are outlined.

Chapter 2 provides an overview of reverse engineering, with the emphasis on the data-acquisition phase. A taxonomy of data acquisition techniques is provided with some discussion of the relative strengths and weaknesses of each method. Common problems in data acquisition are discussed as well as those specific to the type of triangulation sensor used in this project.

Chapter 3 describes a range of image processing techniques that are in common use today. Image processing and computer vision is a huge area of research and it is not possible to cover all aspects within the limits of this chapter. The emphasis is therefore on techniques related to edge-detection methods that are used within the scope of this project.

Chapter 4 outlines how the image analysis methods described will be used to develop the concept of ‘scan orientation regions’ and the new algorithms that will be used to control the sensor in selecting a locally optimal scan orientation as it traverses the object. The currently available hardware environment and modifications required for this project are described. Practical calibration issues concerned with relating the image and scan coordinate systems are discussed and equations that may be applied to compensate for sensor alignment are outlined. A number of systematic issues concerned with the current scanning process are also identified.

Chapter 5 describes the implementation, development and testing of the image analysis aspects of this project. A number of edge detection methods are implemented using a common framework. The construction of straight line edge vectors that may be used as a basis for scan regions is then discussed.

Chapter 6 provides a discussion of the practical implementation of the scan regions method based on edge vectors. Two ‘partial scan’ methods are presented. The scan acquisition process is outlined in order that a scientific approach to reproducing the experiments may be followed. Practical concerns related to the conversion of the scan data into a format suitable for this project are also addressed.

Chapter 7 presents the results, including a detailed discussion of the application of the two scan orientation region methods to a number of objects of different levels of complexity. The results encompass the whole process from edge detection to generation of partial scan path plans evaluation of the resulting scan output.

Chapter 8 provides some ideas for future work that were considered but not implemented due to time constraints on this project.

Chapter 9 discusses the work conducted and the achievements of this research and the conclusions that may be drawn from the work in this thesis.

Finally appendices are provided, containing two conference papers presented by the Author during the course of this thesis and a book chapter submitted for publication. Another appendix contains a detailed mathematical derivation of the algorithm for the calibration of sensor alignment provided to the Author by Dr. Poliakoff.

---

<sup>1</sup> K. Wong, '*Compensation for Distortion in the Imaging Process for 3D Surfaces*', PhD. Thesis, The Nottingham Trent University, 2002.

<sup>2</sup> Toubin M., Laligant O., *et al.*, '*Multiscale Analysis for the Characterization of 3-D Objects*', Proceedings of SPIE, Intelligent Robots and Computer Vision XVII: Algorithms, Techniques and Active Vision, vol. 3522, pp.414-423, October 1998.

<sup>3</sup> Karbacher S. and Haeusler G., '*New Approach for the Modelling and Smoothing of Scattered 3-D Data*', Proceedings of SPIE, Three-Dimensional Image Capture and Applications, vol. 3313, pp.168-177, March 1998.

<sup>4</sup> Shu C. and Xi F., '*Model-based Scanning Path Generation for Inspection*', Proceedings of Second International Conference on 3-D Digital Imaging and Modeling (3DIM '99), pp.118-124, Ottawa, Canada, October 1999.

<sup>5</sup> Lee, S.J. *et al.* '*Laser Scanning Probe With Multiple Detectors Used For Sculptured Surface Digitization In Reverse Engineering*' 2005 J. Phys.: Conf. Ser. 13 pp.155-158

## 2. Reverse Engineering

In order to compete in today's global market companies must find new ways to reduce costs, accelerate product development cycles and improve time-to-market without sacrificing product quality. Conventional product development follows a formal sequence of design, manufacturing, assembly and inspection. In reverse engineering, physical parts are transformed into engineering models and concepts.

Traditionally the source of design data from which products are created was the engineering drawing, as produced by a skilled draughtsman. The development of Computer Aided Design (CAD) software has revolutionised the manufacturing design process. The emergence of powerful CAD packages has allowed engineers to create their designs on computers and output directly as instructions to machine tools. The existence of a computer model provides enormous gains in improving the quality and efficiency of design, manufacture and analysis.

One of the key benefits is the ability to take an existing model and produce a modified version based on the original design. However, there are instances where a mechanical part exists but no computer model, or even paper drawing, exists to re-create the part. This may be because the machine parts were designed before the advent of computers and CAD systems, as well as for parts that were hand-tooled to fit into existing machinery. If such a part breaks it may prove impossible to replace. In other cases it may be necessary to re-engineer an existing part after modification to construct a new improved product or to produce a copy of a part, for which no original drawings or documentation are available.

The advent of modern digitisation methods allows the creation of a CAD model directly from the object. Similarly, an existing part may be used as a prototype and modified within the CAD package, thereby producing a new part [6]. In such cases some means of capturing that information from the physical prototype is required. Reverse engineering typically starts with measuring an existing object so that a model of the surface can be produced for use in a CAD/CAM system[7].

Reverse engineering can trace its roots back to the pantograph, copy lathes and copy mills. These latter systems have the added advantage of storing the detected profile, so that an object can be duplicated many times without repeated scanning [8].

The generation of a CAD model from an experimentally identified prototype is an essential step in the product development process and it is the process of digitising these objects which plays a crucial role in reconstructing their mathematical shape as a computer model.

## **2.1 Other Applications for the 3D Digitisation Process**

In reverse engineering the goal is to recover a design (or produce a new design) based on an existing model. Where that model is a 3D object in the real world the initial step in the reverse engineering process is the shape-capture (or digitisation) of that object. The applications for 3D digitisation stretch beyond the commercial and industrial production of parts based on existing prototypes discussed in the previous section. Some of the other common applications are described here.

**1) Collaborative Design:** CAD tools are often useful in the design process, however in some fields the most intuitive design method is physical interaction with the model. Although computer models provide enormous gains in design quality and efficiency, manufacture and analysis, there are some fields of manufacturing (such as consumer products), where aesthetic design is particularly important. In these areas designers still prefer to evaluate real 3D objects rather than reduced scale 2D projections of those objects on screens. Sculptors design and construct these models using clay or wood and the finalised model is digitised and rendered using graphics software to give an accurate impression of the original model. Engineers or clients access the data model for testing (e.g. finite element analysis) and review. A prototyping, performance-testing and modification cycle is followed to obtain an optimal product design. Once the design is approved it may also provide an initial model for manufacturing. For example, items such as a swimming goggles or football boots may be designed experimentally through conducting actual trials to improve aesthetics and functionality.

**2) Parts Inspection:** The dimensions of the manufactured parts must fall within some tolerances of the original computer model. In this case, shape digitisation can aid in determining where (and to what extent) the computer model and the shape of the actual part differ. These differences can serve as a guide for modifying the manufacturing process until the part production output is acceptable [9].

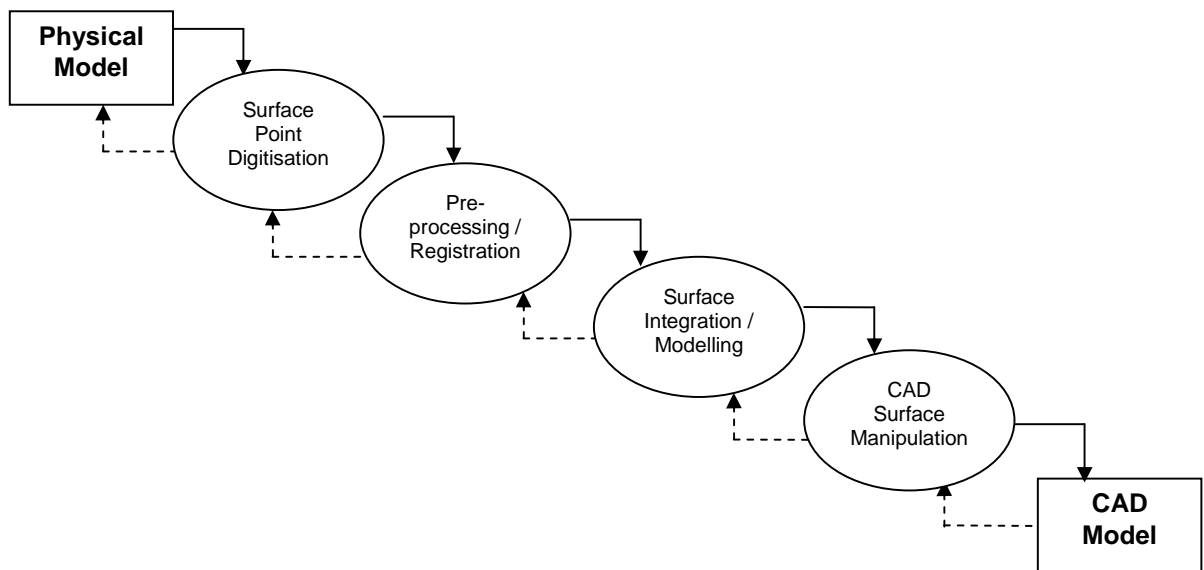
**3) Special Effects, Games and Virtual Environments:** Computer-generated imagery plays an increasingly prominent role in creating special effects for cinema and television. Realistic 3D environments are also a popular part of the computer game industry as well as for more educational purposes such as virtual tours of heritage sites. All of these applications require 3D models that may be taken from real life or from sculptures created by artists. Creating realistic 3D models from scratch is a highly skilled, time-consuming task and the digitisation of physical models, especially for more mundane objects, is essential to the realistic population of these environments [10].

**5) Archaeological Artefacts:** Frequently museums (or private collectors) hold rare objects that are of interest to scientists. In some cases these objects may be too fragile to be examined. Traditionally, in order to visualise these objects, it has been necessary to visit the museum or obtain non-interactive still or video images. By digitising these objects, accurate copies can be made available for interactive visualisation and scientific study [11]. Archaeological site features, architectural elements and sculpture can be digitised to provide a high-resolution 3D digital record of the object or site. Digital models provide archival quality documentation which can be used for a variety of research, conservation, archaeological and architectural applications, fabrication of accurate replicas, as well as for interactive museum displays.

**5) Medicine and Dentistry:** Applications of shape digitisation in medicine are wide ranging. Custom made prosthetics can be designed when the dimensions of the patient are known to high precision. For example, a missing leg can be created using the patient's other leg as a model, and then the digitised model can be mirrored. Plastic surgeons and dentists use the shape of a patient's face to visualise the outcomes of plastic surgery and dental work. In radiation therapy, modelling the patient's body can help guide the doctor in directing the radiation accurately.

## 2.2 The Reverse Engineering Process

The reverse engineering process is comprised of more than just the digitisation of the 3D shape which typically results in the data being represented as individual points. Once the data has been acquired, further processing stages are necessary to produce a model of the object. These stages can be partitioned into a sequence of 4 main steps as shown in Figure 2-1:



**Figure 2-1: Overview of the main steps in the RE process. Note the process is not entirely sequential - steps often overlap and the process may be iterative (as indicated by the dashed arrows)**

1. Data acquisition - Captured data is usually represented by individually sampled points represented by a 3D coordinate 'point cloud'.
2. Registration – To produce an accurate model for true 3D objects a number of point clouds must be combined using a common coordinate system.
3. Integration - Merging surface patches into a single shape representation, creating a smooth, continuous polygon mesh from the registered point cloud data.
4. Optimisation / CAD surface manipulation. - this comprises the 'finishing touches' such as filling any holes in the model surface with a best fit algorithm and any modifications to the model that the user wishes to make.

Within the scope of this thesis the primary concern is with the part of this process referred to as data acquisition. The quality of the chosen data acquisition method can have a great effect on the final quality of the model. It is therefore important to minimise data acquisition errors. The most common techniques for 3D data acquisition are described in the following sections.

### **2.3 Data Acquisition Methods**

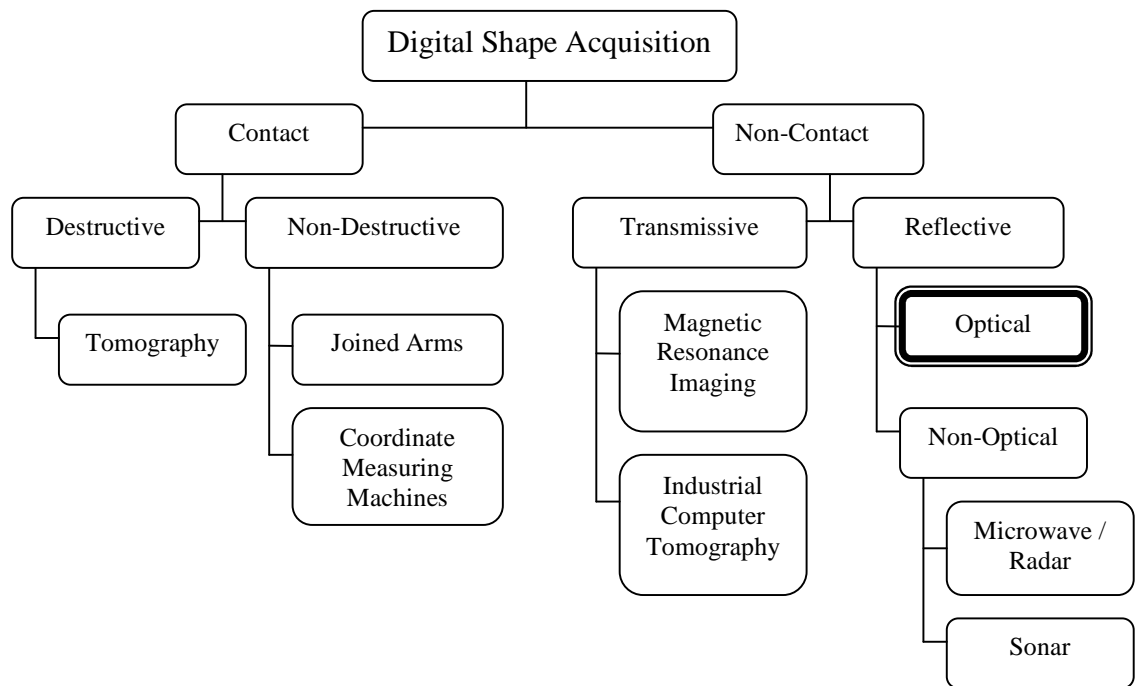
The basis for the reverse engineering model is the capture of the ‘analogue’ real-world model into a digital model. Whether an existing part or a model is used as the prototype, the three-dimensional geometry of the object’s profile must be digitised to produce the design data from which a CAD file can be generated. Many thousands of samples must be acquired, which must then be represented in the CAD model as coordinate points in a 3-D wire-frame view or displayed as rendered surfaces [12].

The data acquisition process, often referred to as 3D scanning or digitisation comprises the measurement and modelling of shape and other visual properties. The accuracy and reliability of the chosen method is an important part of the process. If the method used is not fit for purpose, many of the advantages of reverse engineering the product are negated.

In the last twenty years advances in the fields of photonics and electronic engineering, computer vision and computer graphics have allowed the development of fast, compact, reliable and highly accurate 3D vision systems [13]. The speed and accuracy of these technologies, including the development of lasers and CCDs, enable detailed shape measurements to be made with a level of precision better than 1 part per 1000 and at rates of over 10,000 samples per second [1].

The rapid evolution of computer technologies has been coupled with the development of digital imaging devices over recent years. 3D shape measurement is now at the stage where successful commercial products have been produced based on the techniques proposed in the 1980’s and 90’s.





**Figure 2-2: A Taxonomy of Digital Shape Acquisition Methods.**

Optical methods are further decomposed in Figure 2-4

Numerous methods of data capture have been proposed. Each method has its own particular strengths and weaknesses that make it more or less suited to certain applications. A brief taxonomy of digital shape acquisition methods is illustrated in Figure 2-2.

Early methods for acquiring range images were surveyed by Jarvis [14] in 1983. Besl (1988) [15] surveyed commercially available sensors and measurement techniques. More recent surveys by Tiziani [16] and by Chen [17] describe a wide range of scanning technologies. A broad range of sensors that are commercially available today is presented by Blais [18,19].

### 2.3.1 Tactile Methods

These methods involve physical contact between the surface of the object and the scanning tool. The digitising probe may be a manual or machine-controlled robotic arm equipped with a touch trigger probe as shown in Figure 2-3. Such methods are generally accurate but also tend to be slow and labour intensive [20].

Co-ordinate measuring machines (CMMs) also use a probe and are widely used in industry for dimensional inspection and surface measurement of manufactured parts. Despite the significant improvements in the performance of CMMs over the past decade, several factors (e.g thermal variation) still limit their accuracy, speed, and economic utility. The suitability of these methods also depends on whether the object can be physically touched without causing the object to be distorted (or damaged) during the sampling process.

A more rarely used method is tomography, which involves physically slicing the object into segments of known width and measuring their cross-section in 2 dimensions. The slices are then reassembled in the computer model. This allows internal shape such as cavities to be captured. The primary disadvantage of this method is its obviously destructive nature.



**Figure 2-3: Coordinate Measuring Machine with tactile sensor probe (left) and Manual 'Joined Arms' Measuring System (right).**

### **2.3.2 Non Tactile Methods**

Non-contact methods involve the measurement of some form of energy source. These methods are generally faster than the methods described in Section 2.3.1, however they all have particular advantages and disadvantages relating to the acquisition process, which makes them suitable for particular applications and scenarios. Non-tactile methods can be broadly divided into transmissive and reflective methods.

Transmissive techniques such as computer-tomography (CT) and magnetic resonance imaging (MRI) produce volumetric range images from a series of two-dimensional images [21]. CT scans are commonly used in industry and medicine, where the ability to see the internal surfaces of structures (without damaging the object) is important. Although CT is non-destructive the use of high-energy radiation to produce the scan makes it unsuitable for many applications. The equipment is also very expensive and large variations in material densities (e.g. wood glued to metal) can degrade the accuracy.

MRI systems do not expose the subject to high energy radiation but are even more expensive and are less well suited for industrial applications. These techniques have proved very useful in medical systems where the computed axial tomography (CAT) and MRI scans are used in diagnosing illness and injury without invasive surgery. Optical (visible light) tomographic reconstruction methods have been investigated [22]. This is a similar technique to those using other parts of the electro-magnetic spectrum (e.g. x-rays). Although obviously less harmful, it is limited to transparent objects (e.g. glass and clear plastics).

Reflective techniques can be divided into optical and non-optical approaches. Non-optical approaches include sonar and microwave radar which typically determine distances to objects by measuring the time required for a pulse of energy to bounce back from an object. Microwave radar is typically intended for use with long range remote sensing. Sonar range sensors are typically inexpensive, but they are also not very accurate and do not have a very high acquisition rate. Section 2.3.3 describes the many different reflective optical techniques.

### 2.3.3 Optical Data Acquisition

Generally optical reflective methods are preferred in the field of reverse-engineering due to their higher resolution and acquisition speeds. These optical methods can be divided into active and passive techniques, as shown in Figure 2-4.

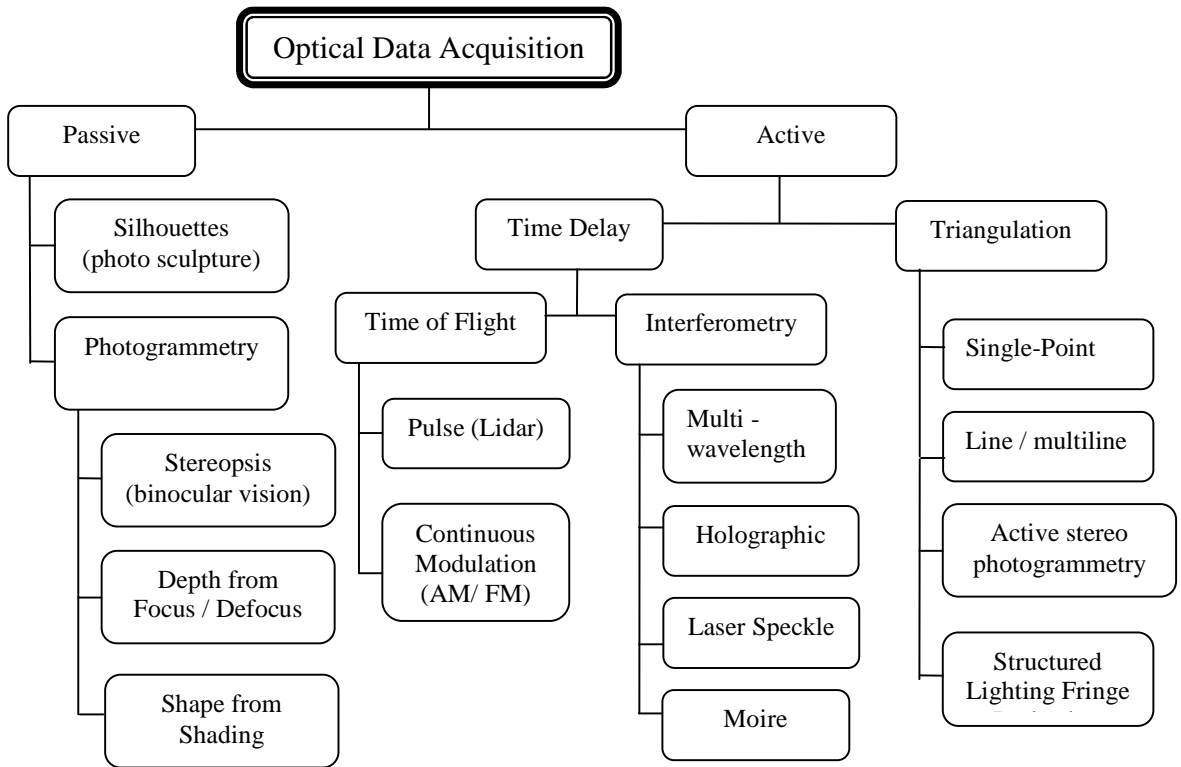


Figure 2-4: A taxonomy of optical data acquisition techniques.

#### 2.3.3.1 Passive Optical Methods

Passive sensing techniques reflect the way human vision works, relying on natural or ambient lighting to illuminate the object. Computer vision approaches include shape-from-shading for single images and optical flow methods for video streams. While these methods require very little special purpose hardware, they typically do not yield dense and highly accurate digitisation required for most applications.

The work described in this thesis employs methods from passive image analysis, specifically the use of edge detection techniques, in order to locate edges in a camera image that correspond to features of the object. However edge detection is not used directly in surface

measurement but as a guide to planning the path of the triangulation sensor (as described in Chapter 4).

**Passive triangulation** encompasses digital photogrammetry, which is widely used in cartography and industrial inspection [23]. Stereophotogrammetry extracts 3D information by matching features (or areas) of the scene taken from different viewpoints [24]. This is a very accurate method (modern systems can achieve accuracy of 1 part in 100,000 or even 1:1,000,000) [25]. However the cost of stereo plotting equipment is relatively high and merging data from multiple views is a non-trivial task. Correlation of image pairs can prove difficult (unless ‘artificial’ reference markers are used) which is why active methods are often preferred.

**Depth from focus** has evolved as both a passive and an active sensing strategy. This method operates on the principle that the image of an object is blurred by an amount proportional to the distance between points on the object and the in-focus object plane. In the passive case, variations in surface reflectance (also called surface texture) are used to determine the amount of blurring. Active methods project a pattern of light (e.g., a checkerboard grid) onto the object. Most prior work in active depth from focus has yielded moderate accuracy (up to one part per 400 over the field of view) [26].

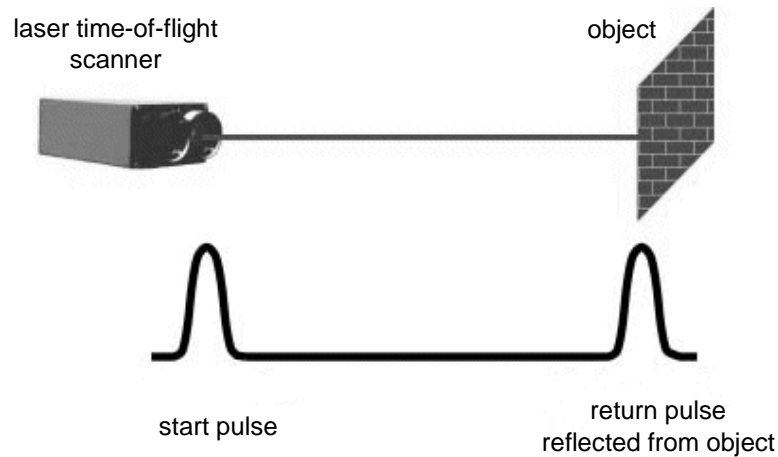
### 2.3.4 Active Optical Methods

Active optical methods project a controlled source of light energy onto the object under consideration. They can be divided into time-delay methods, (encompassing both time-of-flight systems and interferometry methods) and triangulation methods, with light projected in point, line, or multiple-line scan patterns.

Time of Flight (ToF) 3D scanners are the preferred choice for long range measurements [19]. They are relatively simple and accurate systems and a number of variants exist, including pulsed-wave, continuous-wave (AM or FM) systems [27,28].

The accuracy of these sensors is typically limited by the accuracy with which the time interval can be measured. Multiple pulses are averaged to reduce noise and resolution in the order of 0.5 to 1 cm is very common. Higher resolution and accuracy requires very sensitive electronics with high bandwidth. Amplitude modulation of the carrier wave has

also been proposed. The electronics are more complex than pulse ToF, but they give better resolution (in the order of 3 to 5 mm) [19]. Frequency modulation techniques give still better (sub-millimetre) resolution.



**Figure 2-5:** Pulse ToF systems detect the time taken for a laser pulse to be reflected back to a receiving detector [29].

Interferometry, (or modulated-beam imaging), works by determining interference patterns in light wave propagation [30]. Michelson [31] first demonstrated the practical use of optical interferometry. A single light source is split into two or more beams using mirrors [32] which interfere when recombined, to give a wave interference pattern. This pattern appears as a series of bright and dark bands depending on the phase difference between beams. Distances are measured in terms of wavelengths. This can be a very accurate method of measurement but is limited in accuracy by the frequency of modulation and the ability to resolve the phase difference. Interferometers are also highly sensitive to vibration, movement and thermal expansion which further limit their accuracy.

As shown in Figure 2-6, Moiré interferometry involves the projection of coarse, spatially varying light patterns onto the object, whereas holographic methods typically rely on mixing coherent illumination with different wave vectors. Moiré methods can have phase discrimination problems when the surface does not exhibit smooth shape variations.



**Figure 2-6: Moiré Interferometry Pattern**

Coherent holographic interferometers typically yield range accuracy of a fraction of the light wavelength over microscopic fields [33]. Holographic interferometry is often used to visualise stress or to detect object deformations in real time. Optonor offers microscopic vibration measurement systems based on the technique of TV holography (electronic speckle pattern interferometry) [34].

Laser radar 3D imaging, also known as laser speckle pattern sectioning [17] uses the relationship between optical wavelength (frequency) space and the distance (range) in the 3D Fourier transform space to measure the shape of an object.

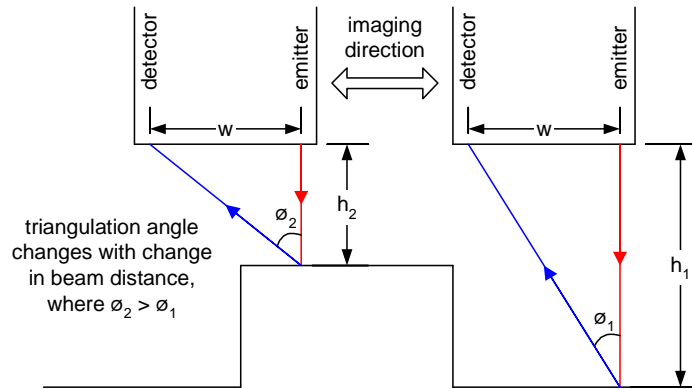
The measurement range can be from a micrometer to a few meters. The accuracy is dependent on the measurement range. With the current laser technology, 1 to 10- $\mu\text{m}$  resolutions are attained in the measurement range of 10 mm with an achievable accuracy of 0.5 $\mu\text{m}$ . The advantages of this technique are the high flexibility of the measurement range and phase shifting as in conventional interferometry may not be required. The limitation of this technique is the time taken to acquire all the different wavelengths images for relatively large scale shape measurement [17].

#### **2.3.4.1 Active Triangulation**

Active triangulation is one of the most common systems used in data acquisition today. Many of the principles, devices and techniques in use today were published during the 1980's, although the theory of triangulation itself dates back to the ancient Greeks who used the principle for navigation and astronomy.

In active triangulation a controlled light source is projected onto the surface of the object. Triangulation sensors are ideal for measuring small distances, fragile parts or soft surfaces susceptible to deformation if touched by a tactile probe.

Figure 2-7 shows an example of how triangulation is used to measure distance using a single point sensor. The emitted light interacts with the surface and is reflected or scattered towards the sensor's detecting element. The receiving sensor is comprised of a light sensitive detector and a focusing lens which concentrates the reflected light onto the detector. The angle between the directions of illumination and detection is known as the *triangulation angle*.



**Figure 2-7: Changes in triangulation angle with changes in distance.**

$h$  is the distance of sensor from spot image,  $\phi$  is triangulation angle and  $w$  is the distance between emitter and detector.

Since the geometrical arrangement of the emitter and the detector is known, the change in the triangulation angle can be used to compute the height. As the light source is moved over the object the position of the detected spot on the detector provides information on the triangulation angle at each sample point.

The formula for height measurement is thus,  $h = \frac{w}{\tan \phi}$

where  $h$  is the distance of sensor from spot image,  $\phi$  is triangulation angle and  $w$  is the distance between emitter and detector.



This calculation depends on the assumption that the light falling on the sensor comes only from the ‘primary’ spot where the beam hits the surface. If this is not the case then distortions are likely to occur. The work described in this thesis involves the single point triangulation method as described in section 2.3.4.5.

#### **2.3.4.2 Structured Light Active Triangulation Techniques**

Structured light techniques involve projecting multiple stripes or patterns of light upon a surface of interest and capturing an image of the resulting pattern as reflected by the surface [35]. The image must then be analyzed to determine coordinates of data points on the surface. These methods include sinusoidal fringe methods such as Moiré [36], and projected coded light binary patterns [37], Gray code pattern [38, 39] or phase-shift information [40,41]. These methods are relatively easy to implement, provide fast, full-field measurement and in certain configurations contain no moving parts as the whole 3D object measured scene is illuminated at one time [42]. With some optimised shape-measurement systems the accuracy achieved is as high as 1 part in 20,000.

#### **2.3.4.3 Single Point Triangulation Laser Sensors**

Specialised applications that require speed, high immunity to ambient light or temperature changes, or an increased depth of field benefit from single point triangulation laser scanners. Figure 2-8 shows the construction of a simple single point triangulation sensor.

Such scanners offer high accuracy and resolution, although they employ point-to-point measurement, so have the drawback of a relatively slow measurement speed when compared to slit scanners. Acquisition of distance information is based on spatial rather than temporal periodicity, and thus the electronics for processing are less complicated and therefore these scanners are more cost-effective.

A number of configurations for single-point lasers have been implemented, with costs varying according to the types of components required. Simple sensors involve moving the laser over the object (or conversely, moving the object under the laser) in a controlled environment. The beam-deflection type of scanner is more expensive because precision galvanometers and large pixel linear CCDs are required. In this type of scanner the laser

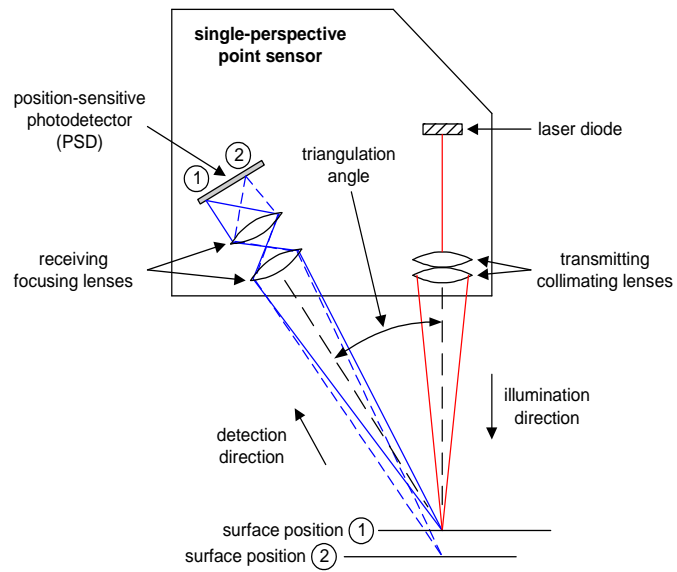
beam is deflected over the surface of the object by the rotation of mirrors, rather than physically moving the sensor position, thus there is less stochastic noise through vibration.

Typical measurement ranges are between 5mm and 250mm with an accuracy of about 1 part in 10,000 and a measurement frequency of 40KHz or more [43]. The position detection of commercial systems is now restricted mostly by physical and optical limits [44]. Figure 2-9 shows a single point laser scanner employing the beam deflection method.

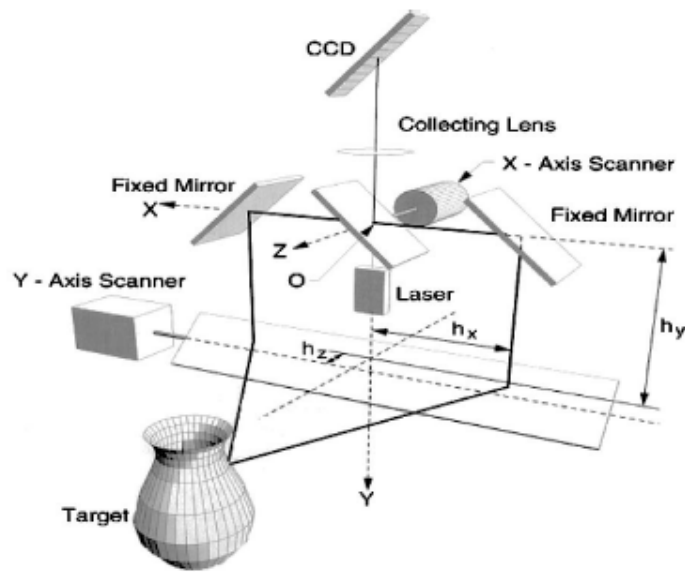
Some single-point laser systems use commercially available laser diode displacement probes with a PSD (position sensitive detector) and analogue signal output, others employ CCD (charge couple device) sensors. For PSD systems the measurement accuracy is dependent mainly on the imaging accuracy of the PSD, which can be affected by stray light and beam spot reflection issues. Idesawa developed some methods to improve the accuracy of the PSD by using a high accuracy kaleidoscopic mirror tunnel position sensing technique (KM-PSM) and a hybrid type of position sensitive detector (R-HPSD) [45].

CCD based sensors avoid the beam spot reflection and stray light effects and provide higher accuracy because of the single pixel resolution. Another factor that affects the measurement accuracy is the difference in the surface characteristic of a measured object from the calibration surface. Usually calibration should be performed on similar surfaces to ensure the measurement accuracy. The recently developed confocal technique can tolerate surface colour change, transparency difference, and irregularity without calibration [46].

Motavalli & Bidanda [47] present a reverse engineering strategy using laser triangulation. Modjarrad [48] presents the use of laser triangulation on a coordinate measuring machine. Physical limits such as speckle noise and resolution are the main limitations, although occlusions, caused by sensor viewing constraints, are also an issue.

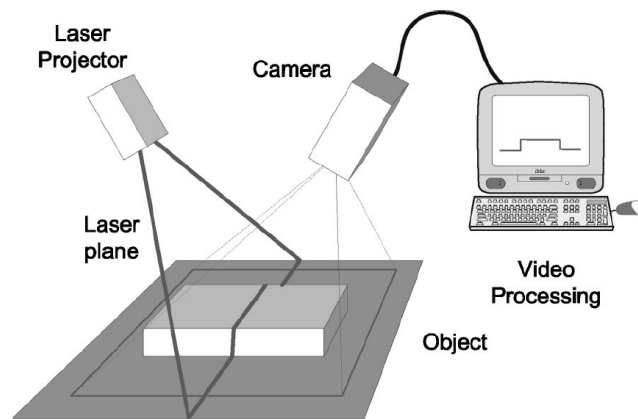


**Figure 2-8: Internal functions of a single-perspective point sensor. (Adapted from Wong [1])**  
The position of the reflected spot on the photodetector moves as surface position changes, allowing the change in distance to be calculated.



**Figure 2-9: Single point laser scanner employing longitudinal synchronisation [19].**

The slit scanner is a linear extension of the single point detector [49]. As illustrated in Figure 2-10, a laser line is projected on the object and imaged at a known angle, allowing the simultaneous detection of a complete profile of points in a single video frame, therefore only one scan, perpendicular to the light stripe, is required for 3-D imaging [50]. The deformation of the profile is a direct function of range [19]. As acquisition of depth information is based on spatial instead of temporal periodicity, the electronics for processing are less complicated and as such they are more cost-effective.



**Figure 2-10: Line (or 'Slit') Laser Scanner.**

Today, slit scanners are the most widely used triangulation-based 3D laser cameras, chiefly because of their relatively low cost. The primary problem associated with this type of scanner is the compromise between depth resolution and field of view which is usually limited to a range of 20 to 30 degrees. A second disadvantage with slit scanners is their relatively poor immunity to ambient light.

## **2.4 Common Problems in Data Acquisition**

As well as system specific considerations there are a number of practical problems with the acquisition of useful data common to all 3D scanning systems in use today. Varady, Cox and Martin [12] recognise a number of major problems in data acquisition: calibration, accuracy, accessibility, occlusions, fixturing, multiple views, noise and incomplete data, surface finish and the statistical distribution of parts.

All methods require calibration as an essential part of setting up and operating a position measuring device in order that parameters such as camera position and orientation are accurately determined so that potential systemic errors (e.g. lens distortions or non-linear electronics) can be modelled.

The key to creating a successful digital reproduction of the object is measuring with sufficient accuracy to capture the degree of detail necessary for a faithful reproduction. Most scanning systems have an optimum working range within which they are most precise in their ability to determine the correct position of the object. Outside of the working range the data is likely to be unreliable. Within the range the level of accuracy is dependent on the type of scanner used. A long range scanner used for measuring the exterior of a building and a close range scanner used for measuring small machine parts may both have an accuracy of one part in ten thousand, but the long range scanner may only be accurate to 10mm while the close range scanner may be accurate to 0.01mm however, accuracy is not the same thing as resolution: a system may have sub-millimetre resolution, but if the detected position is not accurate with respect to the topology of the original surface then the resulting measurement is worthless.

Some parts of the object may be inaccessible with certain types of sensors, due to the topology of the object or physical limitations of the scanner. This problem may sometimes be overcome by making several scans of the object from multiple perspectives but in some cases (for example, through-holes) the correct data may be impossible to acquire. Occlusions occur where the topology of the object causes the scanner to be blocked from detecting the return signal. This is a problem with many optical systems but may also affect acoustic and magnetic scanners.

A common approach is to use multiple scans (or scanning devices) to overcome the problem. The use of multiple views in order to fully describe the shape of a 3D object introduces registration issues between the different scans. In order to recreate the object the whole surface must be reconstructed from only the visible parts.

In many systems, fittings are required to hold the scanned object in the correct location and orientation for the scan. The fittings may cause occlusions and may be included in the data set. Varady *et al.*[12] refer to this issue as ‘fixturing’. Removal of these artefacts from a set of partial scans is a non-trivial task.

Systemic noise can be introduced into the system in many ways, such as vibrations, specular reflections and electrical interference. A number of different filtering approaches may be used, but the filtering or smoothing of data also masks the sharpness of the data.

The surface finish of the part being measured may also introduce noise into the data. The texture of the object, reflective material coatings and even colours can significantly affect the acquisition process. A rough surface will produce more noise than a smooth one. Reflective surfaces may give rise to specular and ‘secondary’ reflections. Coating the object with a matte monochromatic spray or powder to lessen the effect of transitions in reflectivity may be possible in some cases, but then the measured surface also includes the coating, which may obscure fine details and not be distributed evenly.

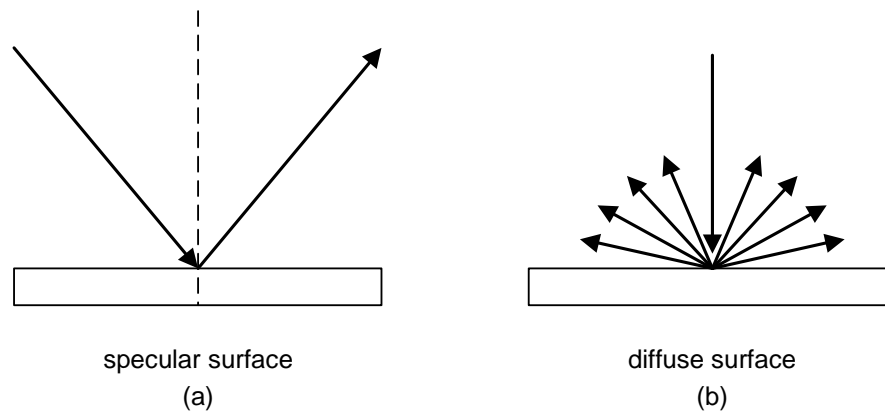
#### **2.4.1 Limitations of Single Point Laser Triangulation Sensors**

Under the right conditions laser point triangulation systems are generally highly accurate and capable of producing results that closely match the profile of the original object. However they are susceptible to a number of recognised issues. Some of these issues are specific to single point lasers and may be solved (or minimised) by the use of more expensive components (such as CCD detectors rather than a PSD); others, such as the occlusion problem, are common to all triangulation methodologies.

### 2.4.1.1 Spot Image Variations and Transitional Changes

There appears to be localisation error from a scattered spot image, known as ‘speckle’ on rough surfaces [51]. The shape of the spot image on the detector depends on the unknown microtopology of the scanned surface [52]. As the spot image is not focused on the detector it yields a noisy, unreliable signal, which is then processed (averaged) by the sensor’s electronics.

The type of material or colour of an object also affects the level of reflectivity from the surface. Contrasting colours affect the intensity of the spot image in transitions from one colour to another. Because triangulation operates by detecting light reflected from the surface, a change in reflectivity affects the level or intensity of light reaching the detector [53]. Abrupt changes in intensity may translate to incorrect readout by the detector as explained below.



**Figure 2-11: Specular surfaces (a) are mirror like, producing a predictable reflection. Diffuse surfaces (b) are irregular, resulting in scattered reflections.**

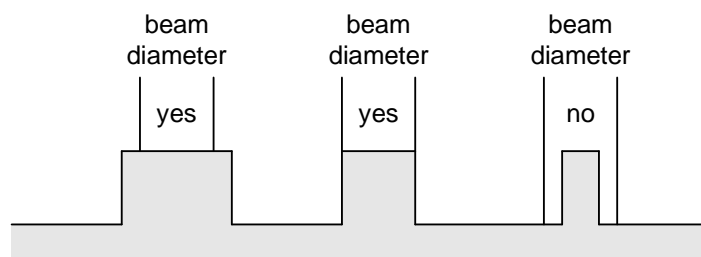
Height measurement is based on the centroidal position of the reflected spot image on the detector. A sudden change in reflectivity causes a shift in the position of the centroid of the spot image on the detector. The spot becomes unevenly distributed as it transitions from a reflective to non-reflective region (or vice versa). Such transitional changes in reflectivity are discussed by Wong [1].

In cases where the surface reflectivity changes dramatically, such as components with different coloured regions, the sensor must be able to respond to these changes automatically. Transparent or translucent objects also yield unreliable depth information, as the spot image penetrates the surface of an object to an unknown depth before being reflected [54].

Applications where this is a factor require a very fast feedback scheme that controls the laser intensity or some other exposure feature in real time to ensure that stable and reliable data is obtained. Therefore the selection of sensor type is dictated by the surface property of the object being examined. A widely adopted approach to mitigating this problem of varying surface reflectivity is to treat the surface of the object with a layer of diffusely reflecting paint to ensure uniform reflectivity.

#### 2.4.1.2 Resolution

The resolution of the system is in part determined by the sensitivity of the detector but the size of the spot dictates the target feature size detection limit. Feature size limitation is the spatial lateral resolution, approximately equal to the spot diameter. When the beam diameter is smaller than or the same size as the feature, the sensor has sufficient resolution. However when the feature is smaller than the beam diameter, the resolution is inadequate for feature detection and measurement. For instance, if the spot diameter is  $50\mu\text{m}$  it will be difficult to resolve a lateral feature of dimensions less than  $50\mu\text{m}$ .

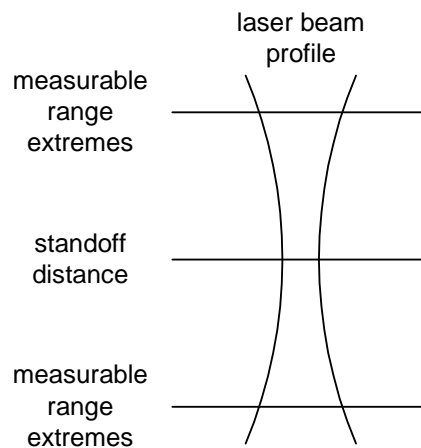


**Figure 2-12: Diameter of beam dictates minimum measurable feature size, or spatial lateral resolution.**

Spot diameter is usually specified in the centre of the measurable range, but the limitations of physical optics dictate that it will not remain at that size throughout the working range.



The collimating lenses form a beam waist that is narrowest at the standoff distance; at the extremes of the measurable range the beam diameter is larger (see Figure 2-13). The rules of feature size detection hold true at the extremes of the measurable range, but as the beam size is larger the smallest detectable feature size is also larger. The limitation imposed by the beam diameter on the feature size that can be detected may be an important consideration for some applications. Another aspect that must be considered relating to spot size is the effect that the spread of the spot has on determining range data as it moves between 2 different heights on the object. Because the spot has a finite width, the transition from between 2 areas of different heights is affected both by the sample rate and the spot diameter.



**Figure 2-13: Schematic diagram of a laser beam. Spot diameter is smallest at the standoff distance. At the range extremes the spot is larger and less precise.**

### **2.4.1.3 Occlusions and Secondary Reflections**

In triangulation, illumination and detection are often not coaxial. Hence it can happen that some part of the object is either not illuminated or the illuminated spot is occluded from the detector's field-of-view. At certain orientations and in proximity to a steep face of an object a single-perspective laser sensor can suffer from occlusions. In such situations the readings at the detector are unreliable and may cause significant distortions.

Distortions due to occlusion are a serious problem with single point perspective scanners and numerous methods have been implemented to mitigate or avoid the issue. Most of these involve the use of multiple detectors around the projector and averaging the signal from each sensor. However these multiple detector lasers are significantly more expensive than a single-detector setup.

Shu and Xi present a method for the automatic generation of scanning paths for 3D line laser scanners based on an existing CAD model [55]. Since normally a single scanning pass cannot cover a whole object without occlusion of some part, a number of scan-passes are performed and combined. Use of this method requires that a CAD model is already present, which is often the case for inspection systems, but sometimes there is no existing computer model.

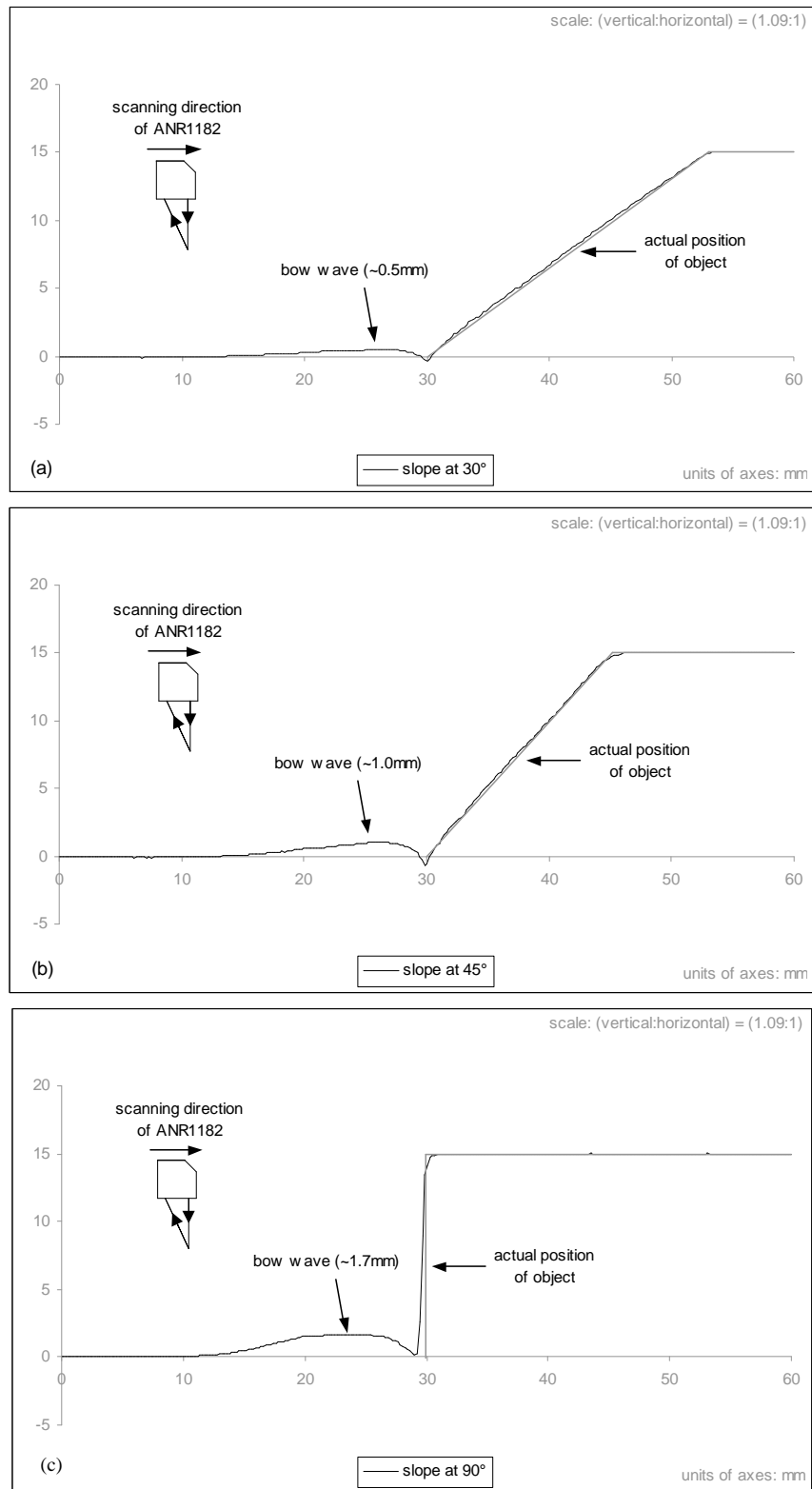
Another drawback of triangulation is that of stray reflections, where the projected spot image on a highly reflective surface results in a small amount of stray light being reflected onto the detector from surfaces that are not directly under the emitter. The combination of such stray reflections with the primary spot image can cause the detector to produce erroneous signals [56]. The relationship between the sensor orientation and significant distortions in the scan data was described in detail by Wong [1]. The salient points with respect to the work discussed in this thesis are summarised in the following section

### **2.4.2 Causes of Occlusion and Secondary Reflection Errors**

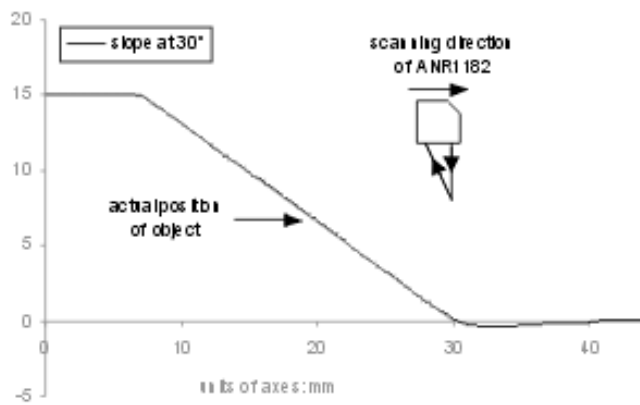
The response of the sensor when it encounters an object 'edge' (a relatively sudden change in gradient or reflectivity) depends mostly on the orientation of the sensor with respect to that edge. The direction of travel is not important – a similar effect would be observed if the laser was crossing the edge in any direction. In the examples shown in Figure 2-15 and 2-16 the scan direction is perpendicular to the edge to show the effect clearly on a single scan line, but the same effect would be apparent by taking a cross section of all scan lines across the region of the edge no matter what direction the scan was made. With the experimental rig the scanning axis employed is the x-axis of the CNC machine. Scans are unidirectional (i.e. the laser scanner is only active on the left-to-right traverse).

It is important to remember that light is reflected in all directions (from an evenly diffuse surface) and it is the effect of the receiving lens that focuses the light falling on the detector window onto the PSD. Some of the light may hit other surfaces and be reflected again, although the reflectance characteristics of the surfaces and the increased distances this secondary light must travel relative to the 'primary' spot will usually significantly weaken its contribution to the overall response, however when the primary laser spot is occluded this stray light has the potential to cause a false reading when there are no stronger signals.

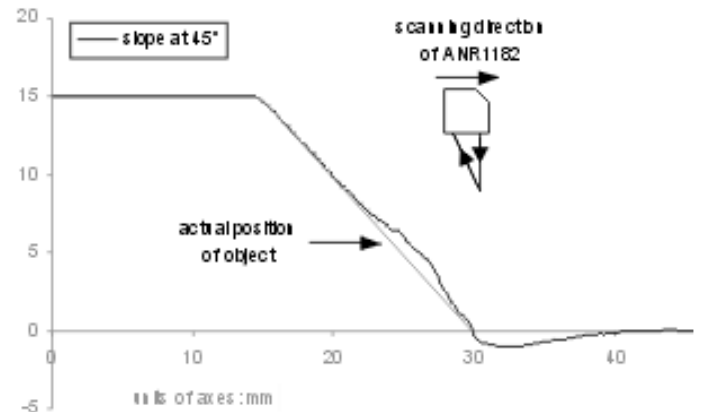
It should be noted that the edge does not have to be (vertically) perpendicular to the sensor in order to cause an error response. As Wong[1] showed, any rapid change in surface gradient has the potential to cause some degree of secondary reflection, and typically object topology may be quite complex, potentially resulting in secondary and possibly even tertiary reflections from multiple surfaces affecting the sensor output. However, the magnitude of both secondary reflections and occlusion errors appear to be at their worst when the edge is vertical, as shown in Figure 2-14 and 2-15.



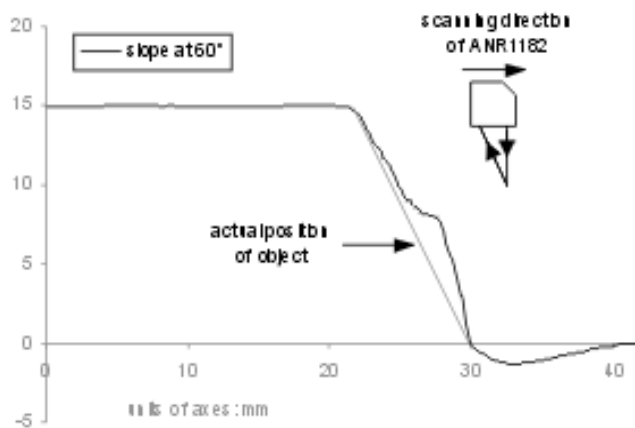
**Figure 2-14: Bow-wave distortions increase with edge gradient (from Wong[1])**



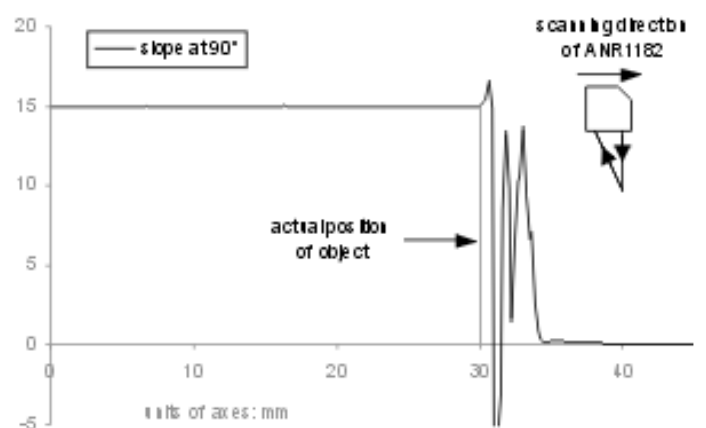
(a)



(b)



(c)



(d)

**2-15: The nature of occlusion spikes changes with the gradient of slope**  
a) 30°, b) 45°, c) 60°, d) 90°. Adapted from Wong [1].

### 2.4.2.1 Occlusion ‘Spikes’

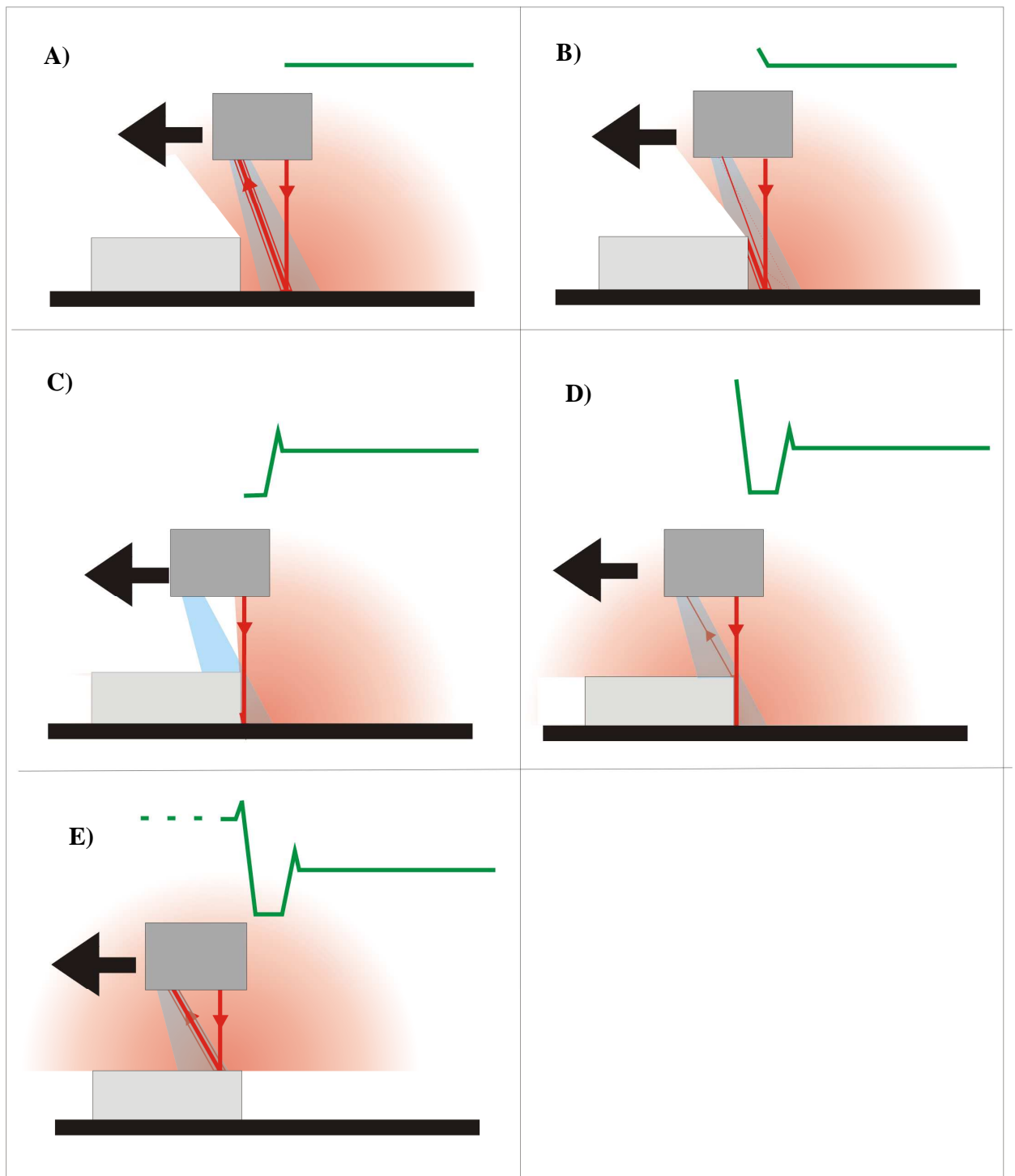
Occlusion of the return laser signal can cause a combination of spikes and troughs to appear along the edge when the return signal is blocked by the geometry of the object (or some other obstruction), thus such errors are generally caused by a relatively steep gradient with respect to the sensor triangulation angle. Occlusion distortions increase in both the magnitude (height distortion) and the width of the effect as the edge height increases. A higher edge intercepts the return signal at a point where the sensor is further from the edge, causing the effect to be wider.

The particular effect (i.e. whether the result is a positive spike or a negative trough) appears to be quite variable, with the relative reflectivity of the surfaces involved and the orientation of the scanner relative to the edge both contributing to the distortion in a somewhat unpredictable fashion, but the effect is usually well localised along the edge.

Figure 2-16 shows an illustration of how the laser samples are affected as the laser approaches a geometric edge with the detector oriented closer to the edge than the emitter, causing the return signal to be occluded. The actual direction of the scan is not important.

At position (A) the laser correctly reads the surface height, but as it reaches the edge (B) the return signal becomes partially blocked. Because the signal spot is spread over a small area, the centroid is blocked but the trailing edge of laser spot is still visible, sometimes causing a slight shift in apparent height (the exact response depends greatly on the relative reflectivity of the surfaces involved) as the centroid of the spot is occluded before the trailing edge.

Secondary reflections from the vertical edge may further complicate the return signal if they fall back on the original surface at a position still within the sensor field of view (i.e. tertiary reflections may also be received at the detector). At position (C) the laser return signal is completely occluded – resulting in an ‘out of range’ signal. Any stray light from reflections hitting the PSD may also cause other false readings. At position (D), as the leading edge of the spot crosses onto the higher surface area, the PSD registers a response again but, since the full width of the spot is not visible, this reading may not be accurate. At position (E) the laser spot is fully on the higher surface and the reading is correct again.



**Figure 2-16: Occlusion errors occur approaching the edge with the detector nearest to the edge. The green line represents the laser response as the detector traverses the edge. The blue area represents the sensor window field of view.**

#### **2.4.2.2 Secondary Reflection ‘Bow Waves’**

The shape of the secondary reflection effect, also referred to as a ‘bow wave’ due to the characteristic shape, is generally less serious than occlusion in terms of the magnitude of the distortion, however the effect tends to be somewhat wider than for the occlusion spikes. Although a vertical edge displays a ‘worst case’ error, any sudden change in gradient can cause some level of secondary reflection.

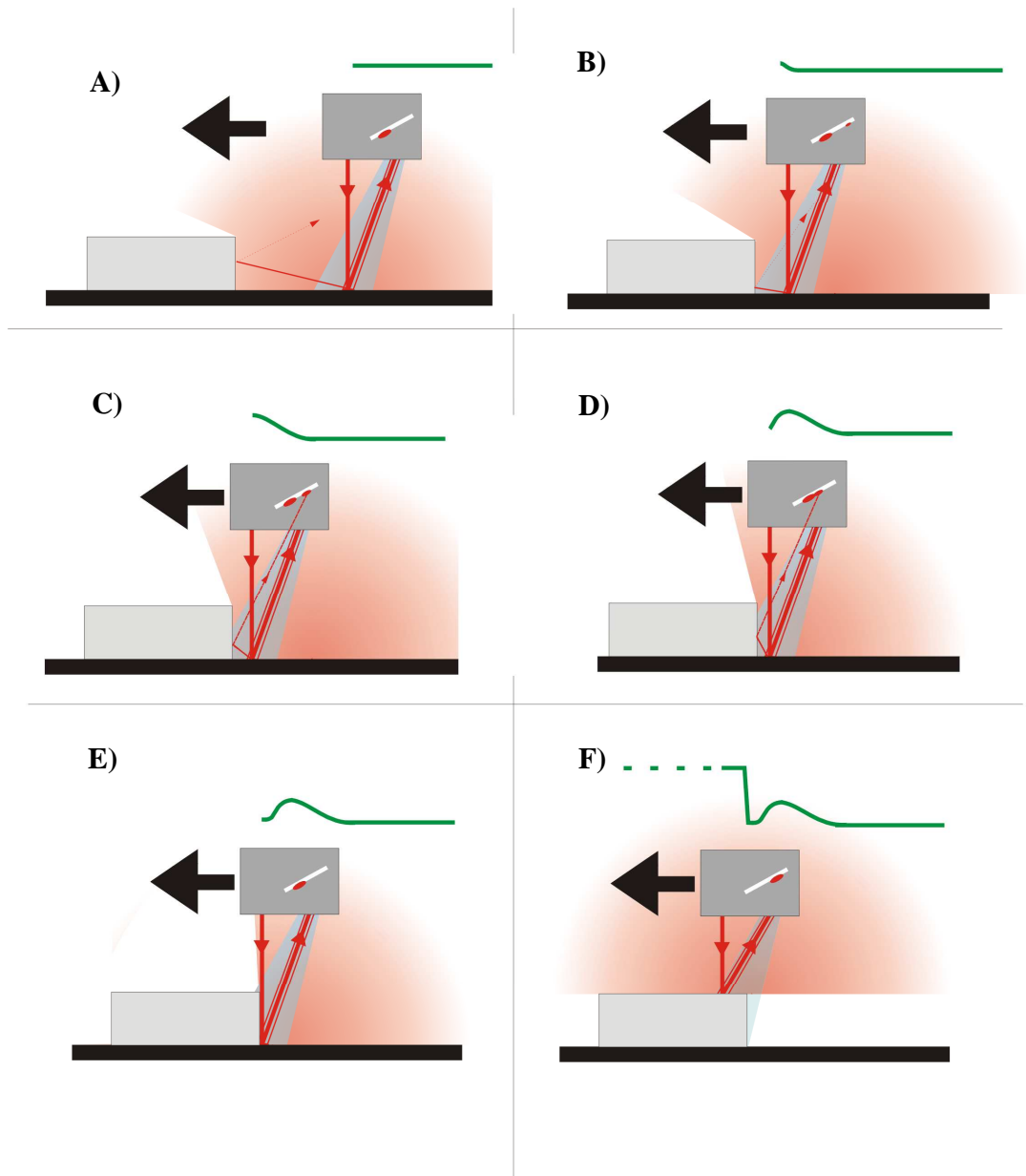
Some of the light from the laser is reflected onto the (vertical) edge surface and some proportion of that incident light will be reflected back towards the sensor. If it falls within the sensor field of view it will be focused onto the PSD. Light may be reflected back onto the original surface, and a percentage of this may again be reflected back into the sensor, so a complex situation of secondary and even tertiary reflections may affect the overall measured spot position. The reflectivity characteristics of the surfaces involved further complicate the observed response.

Figure 2-17 shows an illustration of how the ‘bow wave’ effect may be explained as the laser approaches the edge with the emitter oriented closer to the edge than the detector.

At position A) the laser reads the correct height of the surface. Emitted light hitting the vertical edge surface is reflected back towards the sensor, but the angle of the reflected light is such that the secondary rays are not within the sensor’s field of view, or the increased distance and reflectance characteristics of the edge surface combine to weaken any the signal to a negligible level.

At position B) some of the light reflected from the horizontal surface hits the edge surface within the sensor’s field of view. The light from the secondary reflection must travel further and is partially absorbed and scattered by the edge surface, but some proportion of the light is focused onto the PSD. Because the sensor is relatively far from the edge, the position on the sensor of this secondary spot is some distance from the primary spot, however it is much smaller so it only causes the averaged position of the main spot to be altered by a small amount.





**Figure 2-17: Reflection errors occur approaching the edge with the emitter nearest to the edge. The green line represents the laser response as the detector traverses the edge. The blue area represents the sensor window field of view.**

The relative strength of the secondary reflection increases as the sensor gets closer to the point C), where it reaches a maximum effect, which is a combination of the distance and reflectance of the edge surface. The two spots on the PSD are focused in separate locations, causing the sensor to interpret this as a false average height above the base. As the laser moves still closer to the edge at point D) the secondary reflection still grows in strength, however the separation between the primary and secondary points focused on the PSD decreases, and the average value to become closer to the ‘correct’ surface height again, until the spots converge by point E), where they are interpreted as a single spot. As the laser moves over the edge at point F) the readings revert to the correct interpretation of a single spot.

### **2.4.2.3 Experimental Results**

A series of experiments has been performed to determine the effect of the orientation between the edge and sensor on the scan data. The object used was a small plastic Lego block with a sloped top edge, showing a continuous variation in height from 10mm to 17mm. The block was sprayed with Flawfinder Developer<sup>i</sup> spray to provide a matte response and placed on a neutral grey background. Scans were repeated over the object for laser orientations parallel to the edge and at 9 and 18 degrees, then at further increments of 18°, up to a maximum of 90° (i.e. scanner perpendicular to edge).

Figure 2-18 shows a typical scan-line response curve at two different edge heights (10mm and 16mm) for a number of different laser orientations with respect to the edge. The response curve showed an increasing magnitude and cross-sectional area of the error as the sensor orientation is rotated away from parallel, with the effect increasing both with edge height for both occlusion and bow wave errors.

For both height plots, the left side of each graph shows the effect when the return signal is occluded by the edge of the object. Small deviations from parallel orientation do not cause significant error in either case; however above approximately 18° rotation the typical spike and trough response can easily be seen. At 18° rotation the response is a slight ‘spike’ with no corresponding trough, suggesting that the laser is receiving some reflected light at the

---

<sup>i</sup> Rocol 63135 Flawfinder Developer Spray

PSD at all times, even though the value is incorrect (i.e. the spot briefly becomes partially occluded causing the centroid of the spot to shift on the PSD). For differences in orientation at  $36^\circ$  and above the trough response is indicative of the complete occlusion of the sensor. A small spike precedes the trough in these cases, probably due to the partial occlusion of the spot.

The width of this trough region increases with the sensor is rotated away from parallel. At  $36^\circ$  rotation from parallel the trough width is approximately 2mm where the edge is 10mm high and 2.5mm wide where the edge is 16mm high. This increases to 3.5mm (at 10mm) and 5.5mm (at 16mm) when the sensor is oriented at  $90^\circ$  to the edge. Although the errors are slightly worse for the higher edge, the change in sensor orientation appears to contribute more to the error magnitude than the edge height.

The right side of each graph shows the response of the detector when subjected to secondary reflections from the object edge. The bow wave effect varies with both the height of the edge being measured as well as the laser orientation relative to the edge. The effect characteristically has a 'hump' and a 'tail' area. The 'hump' occurs closer to the edge where the secondary reflections are stronger and then tails off. The height of this region increases with both laser orientation and edge height. Again, the change in the bow wave height caused by changing the orientation outweighs the effect due to edge height.

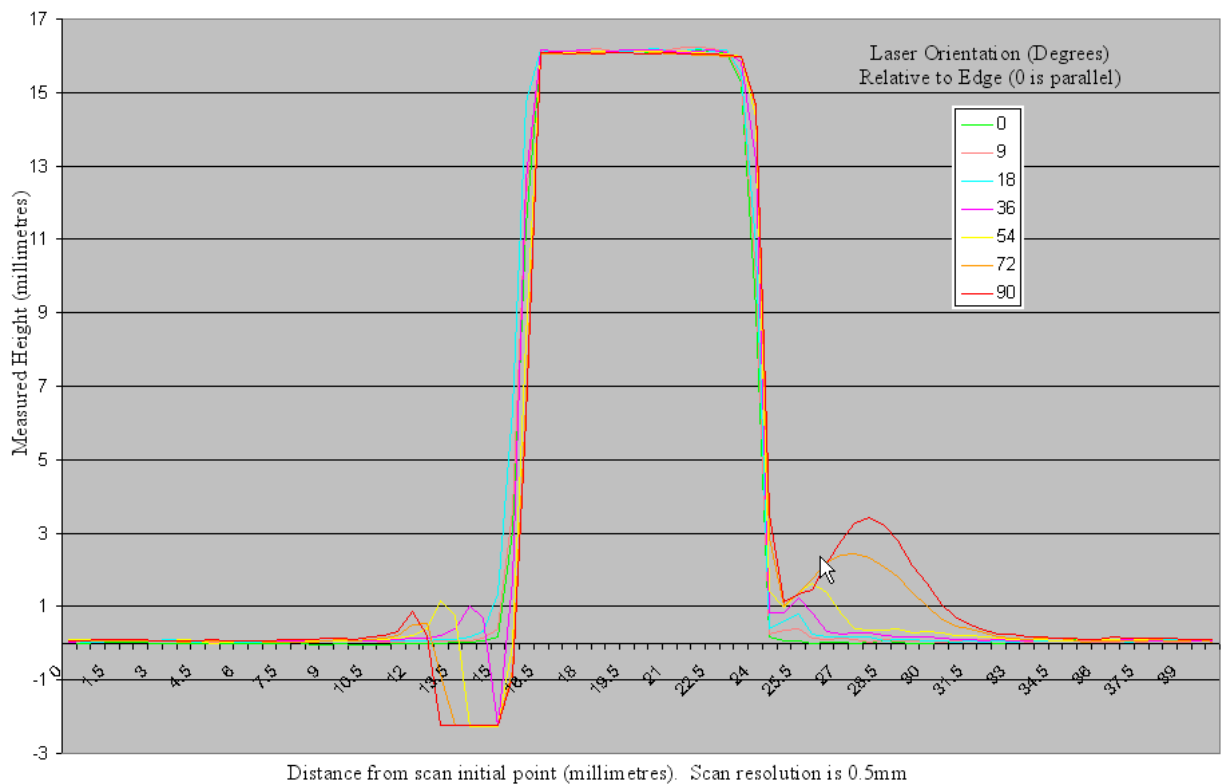
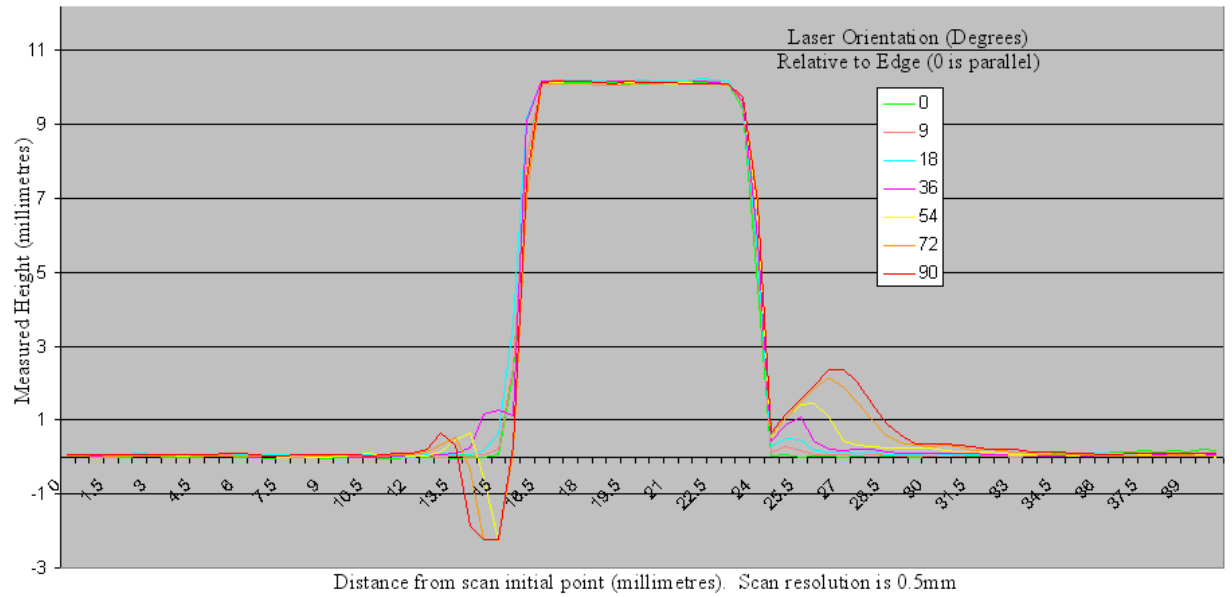
At the 10mm edge height the maximum bow wave height is only 0.5mm from the object edge (as detected by the parallel edge scan) and measures only 0.27mm for a  $9^\circ$  deviation from parallel. This error doubles for an  $18^\circ$  scan, then (approximately) doubles again for a scan with a deviation of  $36^\circ$ . The maximum position of the error also moves away from the 'true' edge position by about 0.5mm for each  $18^\circ$  increment in the orientation. The total width of the bow wave effect for the 10mm edge height ranges from about 4mm at  $9^\circ$  offset up to 9mm when the laser is perpendicular to the edge.

When the edge height is 16mm the maximum bow wave height is 1 mm from the object edge and measures only 0.39mm for a  $9^\circ$  deviation from parallel. This error doubles for an  $18^\circ$  scan and continues to increase as the orientation increases to a maximum of 3.43mm at  $90^\circ$ . The maximum position of the error also moves away from the 'true' edge position as the orientation increases, however the effect does not appear to be as linear as for the lower edge height. When the laser is perpendicular to the edge, the maximum error appears

3.5mm from the edge position, compared to 2mm from the edge where the edge height is 10mm.

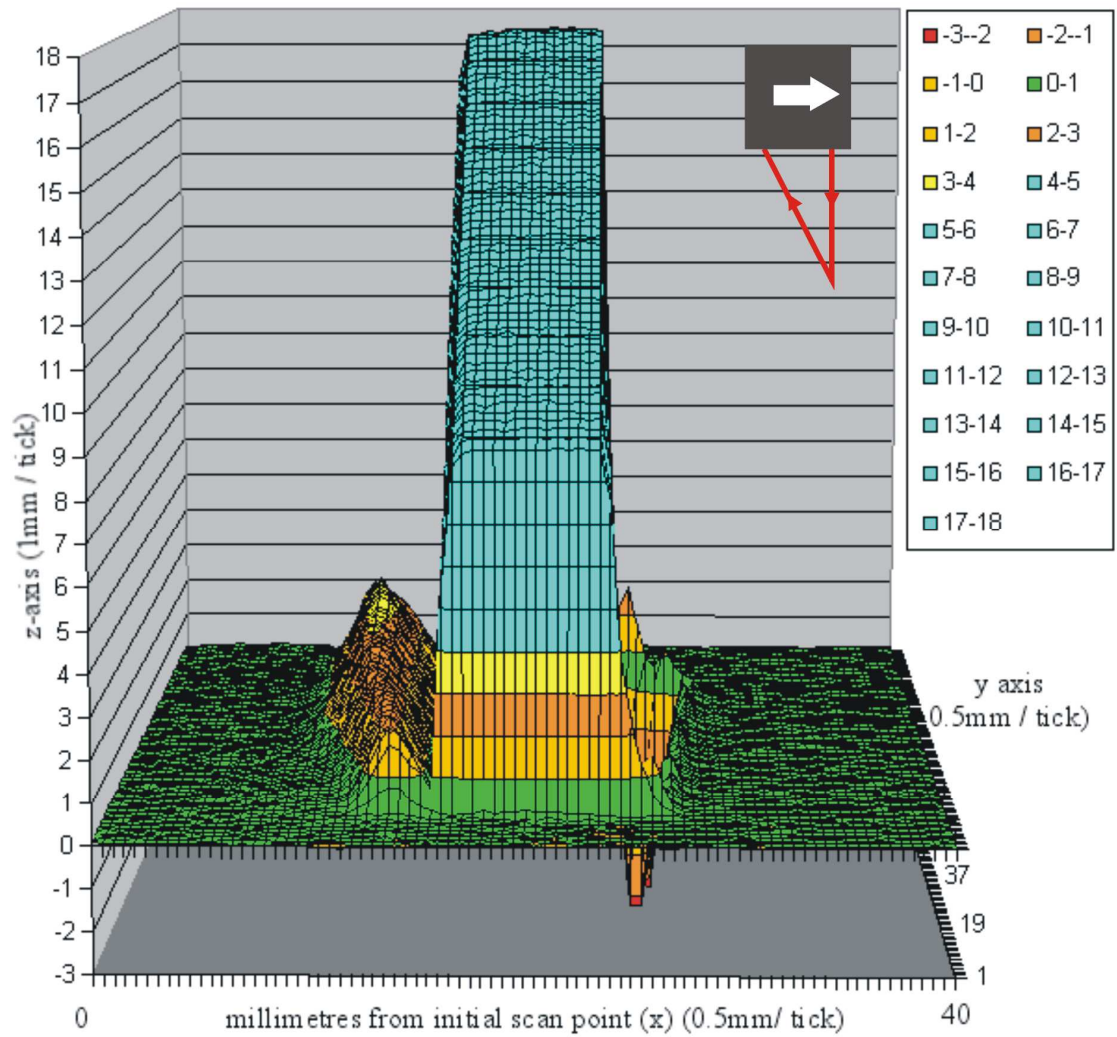
The total width of the bow wave effect for the 16mm edge height ranges from about 4mm at 9° offset up to 11mm when the laser is orthogonal to the edge. Although the effect is wider for a higher edge, the magnitude of the effect appears to be dictated mostly by the orientation of the laser to the edge.

Figure 2-19 shows a complete scan of the Lego block with the laser oriented as in the previous graphs, with the height of the slope increasing along the y-axis. The magnitude of the distortions can be seen to get increasingly higher and wider as the edge height increases. Note that in order to show the slope increasing the graph was rotated 180° from the previous figure (2-18).



**Figure 2-18: Variation in Distortion with Orientation for measured edge heights of 10mm (top) and 16mm (bottom). X and Y scales are consistent between the graphs.**

**The magnitude of both occlusion errors (left edge) and reflection errors (right edge) increase with edge height and as sensor orientation is increased from 0° (parallel) to 90° (perpendicular)**



**Figure 2-19: Captured object shape with laser sensor oriented as indicated. The width and magnitude of both occlusion and bow waves increase with edge height (increasing with the y-axis).**

## **2.5 Summary**

This chapter has discussed how reverse engineering plays a part in the modern product development cycle and some of the main areas in which the methodology has already become established. Many of the methods of data acquisition commonly in use have been described, with special attention given to available active optical methods. The associated advantages and drawbacks for each method are highlighted.

Some problems of data acquisition, such as calibration, resolution and accuracy, occlusion (including fixturing) and systemic noise are common (to some degree) for all methods; other issues are more specific to the particular chosen method.

Basic triangulation sensors suffer from occlusion and reflection errors when the orientation of the scanner is not aligned with the edges. Occlusions occur when the return path of the active sensor signal is blocked by the geometry of the object. Reflection errors occur when the received signal becomes confused by ‘bouncing’ from multiple surfaces.

Multi-perspective sensors still suffer from this problem to a limited extent, but as they include internal methods to recognise that certain sensors are occluded at a given scan position and to exclude the data received from occluded sensors the errors are much reduced, however the financial costs of such sensors is much greater than for single-perspective scanners. Some scanning methods (e.g. structured light methods) can avoid many of these problems but the equipment is often too expensive for small companies.

Within this project the aim is to reduce the errors involved with using a single point active triangulation scanner. The principle types of error related to the use of this method of scanning and their causes have been discussed. The main errors which are of concern in this project are the occlusion and secondary reflection effects seen when a single point optical triangulation scanner (such as the Matsushita ANR-1151) is used.

Previous work in this research group by Wong [1] showed that the primary cause of these errors is the orientation of the laser scanner with respect to the object edge. The dimensions, gradients and relative reflectivity of the surfaces involved also affect the overall sensed value when using a sensor with a PSD. In his thesis Wong attempted to compensate for such errors by scanning the object at a number of orientations (typically 8 scans equally distributed around  $360^\circ$ ) and then identify the 'good quality' parts of each scan by determining the regions of error relative to the edge directions. This method was quite effective but commercially prohibitive because of the total scan time required.

In this project the aim has been to use camera images to identify the position and orientation of object edges prior to the scan process. Then partial scans with optimal sensor orientation may be made and combined, thus saving time by avoiding redundant multiple scans. Chapter 3 discusses how the location and orientation of edges can be discovered through the application of known image analysis techniques.



- 
- <sup>6</sup> Lynch M., '*Computer Numerical Controls for Machining*', McGraw-Hill Professional Publishing, 1992.
- <sup>7</sup> Ingle K. A., '*Reverse Engineering*', McGraw-Hill Professional Publishing, 1994.
- <sup>8</sup> J.C. Owen, P.J. Sloan, and W.B. Thompson, '*Interactive Feature Based Reverse Engineering of Mechanical Parts*', Proceedings of the ARPA Image Understanding Workshop, November, 1994
- <sup>9</sup> Karbacher S., Babst J., and Häusler G., '*Visualization and Detection of Small Defects on Car-Bodies*', Vision, Modeling and Visualization '99, pp. 1-8, Sankt Augustin, Germany, 1999.
- <sup>10</sup> Lassiter J., '*Principles of Traditional Animation Applied to 3-D Computer Animation*', Proceedings of SIGGRAPH '87, pp. 35-44, July 1987.
- <sup>11</sup> Levoy M., '*The Digital Michelangelo Project*', Proceedings of Second International Conference on 3-D Digital Imaging and Modelling (3DIM '99), pp. 2-11, Ottawa, Canada, October 1999.
- <sup>12</sup> Várady T., Martin R. R. and Cox J., '*Reverse Engineering of Geometric Models – An Introduction*', Journal of Computer-Aided Design, vol. 29, no. 4, pp. 255-268, 1997.
- <sup>13</sup> J.-A. Beraldin, F. Blais, L. Cournoyer, M. Rioux, S.H. El-Hakim, R. Rodella, F. Bernier and N. Harrison, '*3D Digital Imaging and Modeling on Remote Sites*', Proc. Second Intern.Conf. on 3D Digital Imaging and Modelling, Ottawa, Canada, (1999).
- <sup>14</sup> R. A. Jarvis, '*A Perspective on Range Finding Techniques for Computer Vision*', IEEE PAMI, Vol. 5, No. 2, 1983 pp 122-139
- <sup>15</sup> Besl, P.J. '*Active, optical range imaging sensors.*' Machine Vision and Applications (1988) vol.1, no.2, p. 127-52.
- <sup>16</sup> H. J. Tiziani, '*Optical metrology of engineering surfaces-scope and trends*' in Optical Measurement Techniques and Applications, P. K. Rastogi, Ed., Artech House, Boston (1997).
- <sup>17</sup> F. Chen, G.M. Brown, M. Song, '*Overview of three-dimensional shape measurement using optical methods*', Opt. Eng., 39, pp. 10-22, Jan. 2000.
- <sup>18</sup> Blais, F. '*Control of low inertia galvanometers for high precision laser scanning systems*', Opt. Eng. 27(2): pp. 104-110; 1988.
- <sup>19</sup> Blais, F. '*Review of 20 Years of Range Sensor Development*', Journal of Electronic Imaging Vol.13 No.1. pp 231-240, Jan. 2004
- <sup>20</sup> Allen P. K. and Michelman P., '*Acquisition and Interpretation of 3-D Sensor Data from Touch*', IEEE Transactions on Robotics and Automation, vol. 6, no. 4, pp. 397-404, August 1990.
- <sup>21</sup> Kak, A. & Slaney, M., '*Principles of Computerized Tomographic Imaging*', 1988, IEEE Press, ISBN 0-87942-198-3
- <sup>22</sup> Trifonov, B, Bradley, D. & Heidrich, W. '*Tomographic Reconstruction of Transparent Objects*', ACM Siggraph 2006, Boston, MA, ISBN 1-59593-366-2

- 
- <sup>23</sup> Fisher P. F. and Lindenber R. E., '*Distinctions among Cartography, Remote Sensing and Geographic Information Systems*', Photogrammetric Engineering and Remote Sensing, vol. 55, no. 10, pp. 1431-1434, 1989.
- <sup>24</sup> Jansa J., Huang Y. R. & Trinder J. C., '*Problems of Precise Target Location and Camera Orientation in Digital Close-Range Photogrammetry*', Proceedings of SPIE, Videometrics II, vol. 2067, pp. 151-161, October 1993.
- <sup>25</sup> Fraser C. S., '*Photogrammetric measurement to one part in a million*', Photogramm. Eng. Remote Sens. vol.58 No.3, pp.305-310 (1992).
- <sup>26</sup> Nayar, S., Watanabe, M. & Noguchi, M. '*Real-time Focus Range Sensor*', Proceedings of IEEE International Conference on Computer Vision, pp. 995-1001, 1995.
- <sup>27</sup> Koskinen M., Kostamovaara J. T. & Myllyla R. A., '*Comparison of Continuous-Wave and Pulsed Time-of-Flight Laser Range-Finding Techniques*', Proceedings of SPIE, Optics, Illumination and Image Sensing for Machine Vision VI, vol. 1614, pp. 296-305, March 1992.
- <sup>28</sup> Moody S. E., '*Commercial Applications of Lidar: Review and Outlook*', Proceedings of SPIE, Optical Remote Sensing for Industry and Environmental Monitoring, vol. 3504, pp. 41-44, August 1998.
- <sup>29</sup> Moring I., Ailisto H., Koivunen V., & Myllyla, R. '*Active 3-D vision system for automatic model-based shape inspection*', Opt. Lasers Eng. 10, pp. 3-4, 1989.
- <sup>30</sup> Lihachev D. I., '*Prospects of Laser Interferometers*', Proceedings of SPIE, International Conference on Lasers for Measurements and Information Transfer, vol. 4316, pp. 43-45, November 2000.
- <sup>31</sup> Michelson A. A., '*On the Application of Interference Methods to Astronomical Measurements*', Philosophical Magazine and Journal of Science, vol. 30, pp. 1-20, July 1890.
- <sup>32</sup> Xue M., Xiangli B. and An B., '*Optical Systems of Imaging Interferometers*', Proceedings of SPIE, International Optical Design Conference, vol. 3482, pp. 474-483, September 1998.
- <sup>33</sup> Curless, B. '*New Methods for Surface Reconstruction from Range Images*'. Thesis published 1997, Stanford University, USA
- <sup>34</sup> Optonor website <http://www.optonor.no/SNT.htm> (accessed 30/08/06)
- <sup>35</sup> K. Harding and L. Bieman, '*High speed moiré contouring methods analysis*', Proc. SPIE 3520, pp. 27-35 (1998).
- <sup>36</sup> Harthong J., Becker A., '*Inverse Moiré*', Proceedings of SPIE, Optical Inspection and Micromasurements II, vol. 3098, pp. 2-9, September 1997.
- <sup>37</sup> Altschuler M. D., Altschuler B. R. and Taboda J., '*Laser Electro-Optic System for Rapid 3-D Topographic Mapping of Surfaces*', Optical Engineering, vol. 20, no. 6, pp. 953-961, December 1981.
- <sup>38</sup> Sato K. and Inokuchi S., '*Three-Dimensional Surface Measurement by Space Encoding Range Imaging*', Journal of Robotic Systems, vol. 2, no. 1, pp. 27-39, 1985.
- <sup>39</sup> Reichmann W., '*Fast Object Recording by means of Structured Light and Photogrammetric Techniques*', Proceedings of IAPRS, vol. 30, pp. 195-200, 1995.
- <sup>40</sup> Srinivasan V., Liu H. C. and Halioua M., '*Automated Phase-Measuring Profilometry: A Phase Mapping Approach*', Applied Optics, vol. 24, no. 2, pp. 185-188, 1985.

- 
- <sup>41</sup> Fang Q. and Zheng S., '*Linearly Coded Profilometry*', Applied Optics, vol. 36, no. 11, pp. 2401-2407, 1997.
- <sup>42</sup> Gaertner H., Lehle P., Tiziani H. J. & Voland C., '*Structured Light Measurement by Double Scan Technique*', Proceedings of SPIE, Vision Systems: Sensors, Sensor Systems and Components, vol. 2784, pp. 21-30, August 1996.
- <sup>43</sup> Keferstein C. P., and Marxer, M. '*Testing Bench for Laser Triangulation Sensors*', Sens. Rev. vol 18 no. 3, pp. 183-187. 1998
- <sup>44</sup> W. J. Smith, *Optical Engineering*, 2nd ed., McGraw-Hill, New York, 1990
- <sup>45</sup> Idesawa, M. '*High-precision image position sensing methods suitable for 3-D measurement*', Opt. Lasers Eng. 10, 3-4 (1989).
- <sup>46</sup> Keyence Technical Report on Sensors and Measuring Instruments (1997).
- <sup>47</sup> S. Motavalli, B. Bidanda, '*A Part Image Reconstruction System for Reverse Engineering of Design Modifications*', Journal of Manufacturing Systems , Vol. 10, No. 5, 1991, pp 383-395
- <sup>48</sup> A. Modjarrad, '*Non-Contact Measurement Using a Laser Scanning Probe*', SPIE, Vol. 1012 In-Process Optical Measurements, 1988, pp 229-239
- <sup>49</sup> Rioux M., '*Laser Range Finder based on Synchronised Scanners*', Applied Optics, vol. 23, no. 21, pp. 3837-3843, 1984.
- <sup>50</sup> de Bakker M., Verbeek P. W. & Steenvoorden G. K., '*Smart PSD Array for Sheet-of-Light Range Imaging*', Proceedings of SPIE, vol. 3965, pp. 21-32, May 2000.
- <sup>51</sup> Häusler G. & Herrmann J., '*Physical Limits of 3-D Sensing*', Proceedings of SPIE, Optics, Illumination and Image Sensing for Machine Vision VII, vol. 1822, pp. 150-158, March 1993.
- <sup>52</sup> Häusler G., '*Three-Dimensional Sensors – Potentials and Limitations*', Handbook of Computer Vision and Applications, Academic Press, vol. 1, no. 19, pp. 485-506, 1999.
- <sup>53</sup> Wolff L. B., '*Relative Brightness of Specular and Diffuse Reflection*', Optical Engineering, vol. 33, no. 1, pp. 285-293, January 1994.
- <sup>54</sup> Dorsch R., Herrmann J. and Häusler G., '*Laser Triangulation: Fundamental Uncertainty of Distance Measurement*', Applied Optics, vol. 33, no. 7, pp. 1306-1314, 1994.
- <sup>55</sup> Shu C. and Xi F., '*Model-based Scanning Path Generation for Inspection*', Proceedings of Second International Conference on 3-D Digital Imaging and Modeling (3DIM '99), pp. 118-124, Ottawa, Canada, October 1999
- <sup>56</sup> Amann M. C., Bosch T. M., et al., '*Laser Ranging: A Critical Review of Unusual Techniques for Distance Measurement*', Optical Engineering, vol. 40, no. 1, January 2001.

## Chapter 3 - Image Processing

Ballard and Brown [57] define computer vision as the enterprise of automating and integrating a range of processes and representations for visual perception. Image processing and analysis techniques are a part of the computer vision field concerned with the enhancement, detection and recognition of features within a scene captured by a still or video camera.

The aim of this project is to use such an image to determine information about the geometry of an object prior to scanning. The field of computer vision contains a vast body of work and a résumé of all the methods is beyond the scope of this thesis. This chapter provides a brief overview of the techniques used in this project. The main themes of thresholding, edge detection, feature extraction (lines and corners) are presented. More detail can be found in many image processing textbooks (e.g. *Digital Image Processing* [58]) and online resources such as The Computer Vision Homepage [59].

Image segmentation is the process of identifying discontinuities and coherent regions in images that (hopefully) correspond to ‘real’ objects. It is recognised that image segmentation is a non-trivial task for any ‘real world’ image. There are three major reasons why this is so difficult. Firstly, a great deal of information (e.g. depth information) is lost when 3D scenes are projected in a 2D image and foreground objects may partially obscure (occlude) other objects. Secondly, segmentation attempts to produce object region primitives; however is hard to define what exactly constitutes a primitive object. Perhaps most importantly, humans use our brains extensively in our perceptual processes. We easily recognise that certain parts belong (or do not belong) together from our knowledge and world experience. Two main approaches have been the mainstay of image segmentation research: region-based and edge-based methods [60]. Each technique has strengths and weaknesses and literally hundreds of papers have been published that propose variations based on these approaches.

### **3.1 Image Kernels and Convolution**

Image kernels are widely- used in image processing to produce filtering effects. A kernel is an array of values that is moved stepwise over the image. At each step, each pixel in the image window under the kernel is associated with the corresponding position in the kernel. Typically each template element is multiplied by the corresponding image pixel value and the sum of the results is recorded in a new output image. This operation is actually the cross-correlation, however the term ‘convolution’ has been loosely interpreted to mean cross-correlation and most image-processing literature refers to this operation rather than true convolution [61].

### **3.2 Image Smoothing**

Image smoothing is often carried out prior to edge or region detection in order to regularise the image differentiation. Generally smoothing employs a low-pass filter, causing high frequency edges to be suppressed and reducing the variation of intensity between neighbouring pixels. Usually the image is smoothed in the spatial domain, although it can also be performed in the frequency domain.

Typical smoothing masks perform an averaging process (returning either the mean or median of the values covered by the mask area). Non-linear filtering has been shown to be more successful than linear filtering because it is better at removing noise whilst preserving edge information [62].

Gaussian smoothing [63] is similar, however, the Gaussian filter outputs a weighted average of each pixel’s neighbourhood, favouring the central pixels’ values, so providing a gentler smoothing and better edge preservation than a mean filter. The degree of smoothing is determined by the standard deviation of the Gaussian.

There are undesirable effects associated with smoothing, such as the loss of information and displacement of structures, and therefore the use of image smoothing must be used with some degree of caution. Large values for the Gaussian  $\sigma$  (sigma) parameter also cause a loss of image data around edges. In order to avoid such loss of data the Green function can be used [64]. A disadvantage of both the Gaussian and the Green functions is that they require the selection of a control parameter by the user / operator.

### **3.3 Edge Detection**

In computer vision, edge detection is a process which attempts to capture the significant features of objects in the image represented by discontinuities in their photometric properties. These features appear as variations in the image intensity such as step and line edges and junctions and edge detection aims to identify the positions of these features. Edges characterise object boundaries and are therefore useful for segmentation, registration and identification of objects in scenes [65].

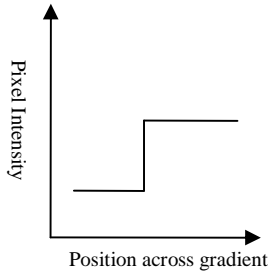
Edge detectors take discrete, digitised images as input and produce output in the form of an edge map. Some detectors provide information only about the location of edges, others provide information about the relative edge strength, orientation and scale. Most edge detectors apply some method to measure the intensity gradient at each point in the image.

Many methods of edge detection have been proposed and most share a number of common features and comprise three main steps: smoothing, differentiation and labelling.

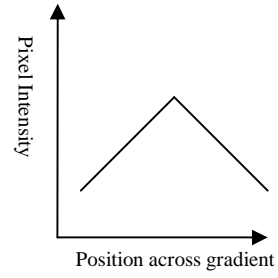
Surveys of edge detection methods may be found in Poggio & Torre [57], Nalwa[66], Zamperoni[67] and Heath *et al.*[79]. Other methods have been investigated by Russo [68], Kundu [69] and Bovik *et al.* [70]. Successful, reliable detection of image edges remains a problem in computer vision, as no single edge detection algorithm is equally successful over all images and situations.

#### **3.3.1 Definition of an Image Edge**

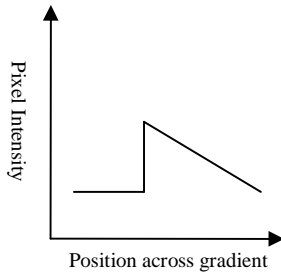
Object boundaries in an image are typically characterised by edges, consequently edge detection is a useful tool in segmenting an image. An edge in a 2D image may be defined as a discontinuity or abrupt change in grey level (intensity) or colour. Edges usually correspond to significant variation in reflectance, illumination, orientation and depth of surfaces and are typically associated with photometric, geometric and physical characteristics of objects within the image [71]. Common types of variation in image intensity are shown in Fig 3-1. Of these, step edges are usually the most commonly encountered type of edge.



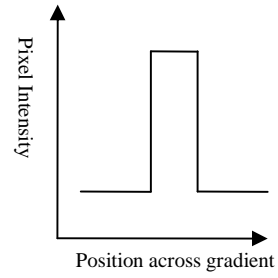
**a) Ideal 'Step' Edge**



**b) Ideal 'Roof' Edge**



**c) 'Ramp' Edges – a combination of step & roof edges**



**d) Ideal Spike or 'Ridge' Edge**

**Figure 3-1: Idealised edge response types. In practice edges in 'real world' images are not often represented by such 'perfect' discontinuities, making the task of edge detection much more difficult than it first appears.**

### 3.3.2 First Derivative (Gradient) Edge Detectors

The magnitude of the first derivative is obtained by performing a matrix convolution of the image with a filter or 'mask'. Ideally, small-sized masks are employed in order to detect fine variation in grey level distribution (i.e. micro-edges). On the other hand, large-sized masks are required in order to detect coarse variation in grey level distribution (i.e. macro-edges) and to filter-out noise and other irregularities. The Sobel [72] and Prewitt [73] masks are popular first derivative edge detectors. The gradient magnitude is given by:

$$|G| = \sqrt{G_x^2 + G_y^2}$$

where  $G_x$  and  $G_y$  is the response of the respective mask when convolved with the image.

One advantage of using first derivative detectors is that it is possible to derive the edge orientation from the edge magnitude information, as given by:

$$\theta = \arctan (G_y/G_x)$$

For this project, the orientation of the (camera) image relative to the laser is calibrated so this edge orientation information can be used to determine the optimal scan orientation for the laser sensor at any given point. However this information is very susceptible to local fluctuations.

Compass variants provide a set of convolution masks which measure the gradient in a selected number of directions, using set of edge templates, each representing an edge at a known orientation. The edge magnitude and orientation then determined by the template that best matches the local area of each pixel. Examples include the Kirsch and Prewitt compass operators. Compass operators require 8 convolutions for each pixel (although symmetrical masks e.g. Prewitt only require four), whereas gradient operators needs only two convolutions. The result for the edge magnitude image is very similar for both gradient and compass methods, provided that the same kernel is used.

### 3.3.3 The Canny Edge Detector

The Canny algorithm [74] is a very popular method which was designed to be an optimal edge detector and arose originally from work in modelling the early stages of human visual perception. It is commonly used in various applications and generally accepted as a *de facto* standard to which many new algorithms are compared. In his paper Canny defined three criteria that edge detection algorithms should meet:

1. Low error percentage
2. Good localisation
3. One response per edge

Deriche [75] extended Canny's initial filter to 2D using recursive filtering.

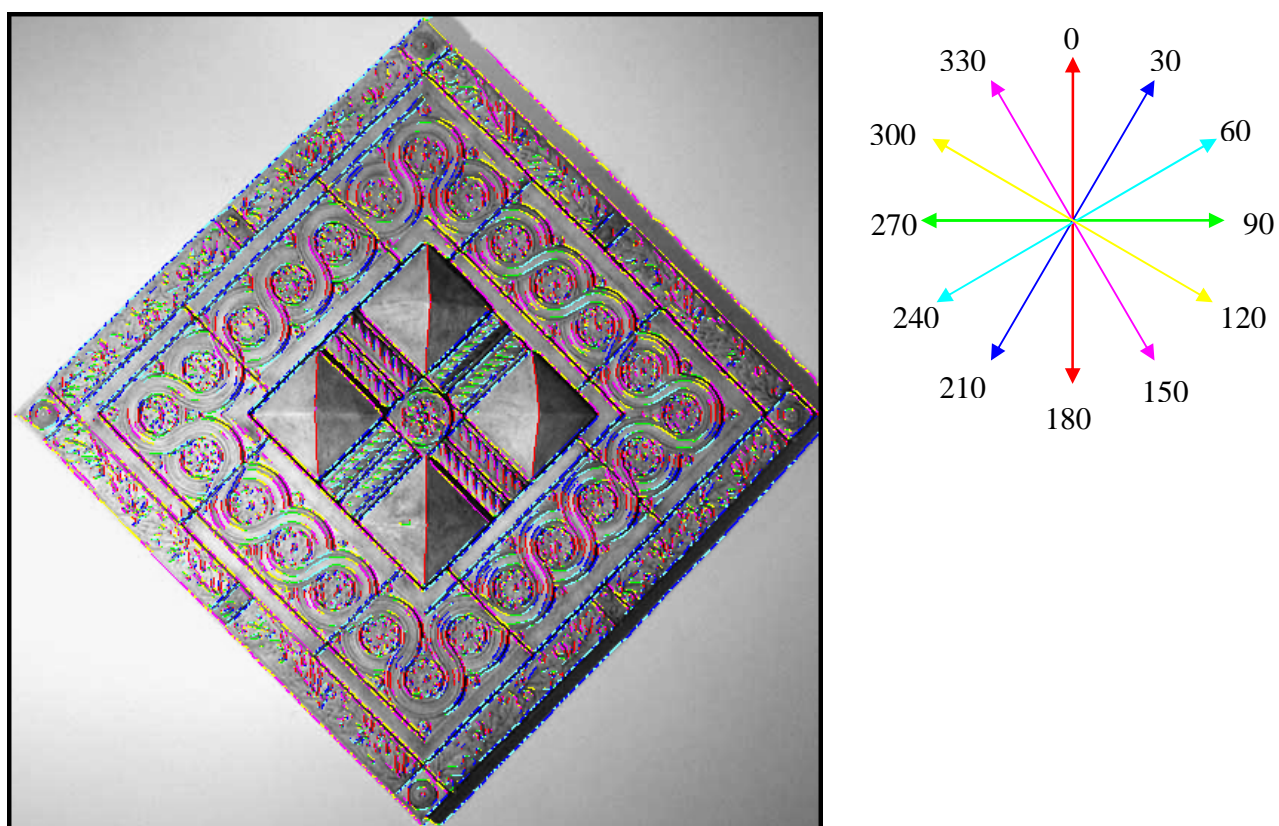
The Canny algorithm is a multi-stage process described in brief here:

- 1) Smooth the image (using a Gaussian filter).
- 2) Compute gradient magnitude and direction (using a first derivative detector).
- 3) Localise edges by non-maximal suppression and perform hysteresis thresholding.
- 4) Repeat steps 1-4 for different levels of smoothing (scale).

(In many cases this last step is omitted from the process).



The effect of the Canny operator is determined by three parameters: the width of the Gaussian mask used in the smoothing phase, and the upper and lower thresholds used by the edge-tracking and hysteresis phase. One problem with the Canny operator occurs at junctions *i.e.* places where two or more edges meet in the gradient magnitude image. The edge tracking algorithm chooses to follow one of the two possible edges at the junction, producing a single line segment, the other edge will appear as a line that approaches (but doesn't quite connect with) that line segment due to the non-maximal suppression effect. This problem can be solved by including a model of such junctions in the edge-localisation phase [76].



**Figure 3-2: Tile image showing edges detected using Canny method. Detected edges have been overlaid back onto the original image to show localisation. False colouring provides an indication of the orientation of the edges according to the key shown (right).**

### 3.3.4 Second Derivative Detectors

Second order derivatives can be used to detect the position of maximum change in the gradient (e.g. the Laplacian function). The Laplacian operator is particularly useful when the grey level transition at the edge is gradual, however it suffers from the drawbacks of second derivative operators, being extremely sensitive to noise. Also it cannot detect edge orientation and may respond doubly to some edges.

Second derivative masks can be combined with gaussian smoothing (i.e. Laplacian of Gaussian) in order to reduce their sensitivity to noise. The zero crossings of the second derivative (Marr and Hildreth [77]) provide a useful way of locating edges. The output from the zero-crossing detector is usually a binary image with single-pixel thickness lines indicating the positions of the zero-crossing points. The results are strongly influenced by the size of the Gaussian used. As the smoothing level is increased then more of the smaller features will be smoothed out of existence. It is possible to approximate the LoG filter with a filter that is just the difference of two differently sized Gaussians. Such a filter is known as a 'Difference of Gaussians' (DoG) filter. Marr [78] used DoG filters in biological visual processing.

### 3.3.5 Assessment of Edge Detection Algorithms

Edge detectors are notoriously difficult to assess in terms of their performance. Purely quantitative statistical assessment can prove to be misleading. The detection of edges falls into 3 basic categories:

1. 'true positive' - an edge is detected at the correct location in the image. Also a 'true negative' result occurs when no edge is detected where no edges exist).
2. 'false positive' - an edge is detected where no apparent edge exists in the image.
3. 'false negative' - an edge in the image is missed by the detector.

Additionally, the edge localisation may be insufficient to accurately represent the object. An edge detector that performs well in a general context produces edge primitives from which an object may be identified. Despite major research efforts, edge detectors do not meet the requirements of many applications in the field of computer vision due to false positives, false negatives and edge localisation errors. These errors depend on image characteristics and properties of the detector.

Heath *et al.* [79] introduce a methodology for rating the output of low-level vision algorithms by humans based on methods borrowed from psychology and statistics. They suggest that each of the tested edge detectors is best for a certain type of image, which is contrary to the assumption that edge detection is purely a context independent process. Heath *et al.* recommend that if the images to be analysed are all fairly similar in content, or if the application allows for tuning the parameters of the detector on a per image basis, then the best choice is a well-tuned Canny detector.

The evaluation of edge detectors is usually performed by human observers, resulting in a subjective assessment [80, 82]. Most of the objective evaluation methods assume knowledge of specific features, such as known object boundaries in simple synthetic images. In such simple cases edge detection can be measured quantitatively, based on the known ‘ideal’ edge detection considered to be the *Ground Truth* [81]. Most implementations of edge detection algorithms involve the *a priori* selection of one or more parameters. There is no automatic parameter selection process that will work in all cases, the optimal parameter set depends on the input image and it is difficult to assess the results of the algorithms for ‘real-world’ images [82].

### 3.3.6 Corner Feature Detection

Corners as well as edges are required for a full image description. Methods using binary edge maps to find corners have been suggested [83]. Beaudet [84] enhanced high curvature edges by calculating image Gaussian curvature. Moravec [85] developed ‘Points of Interest’, where large intensity variations exist in every direction. Harris and Stephens [86] described what has become known as the Plessey detector, built on similar ideas to the Moravec operator. Zheng *et al.* [87] proposed a gradient-direction corner detector that was developed from the Plessey corner detector. Kitchen and Rosenfeld [88] used a local quadratic fit to find corners. Wang and Binford [89] created a detector insensitive to shading. In practice most corner detectors are usually not very robust and often require expert supervision to prevent the effect of individual errors from dominating the recognition task. Smith and Brady [90] proposed the Smallest Univalued Segment Assimilating Nucleus (SUSAN) corner detector which is insensitive to noise and is very fast as only simple operations are used. It performs well in inherently noisy ‘real world’ images because it does not rely on image derivatives.

### **3.4 Thresholding**

Thresholding is considered one of the most basic image segmentation techniques. Each pixel in the image is compared to one or more defined ‘threshold values’ in order to determine to which class it belongs. The effect is to remove grey-level trends within the image, making regions more discrete and segmented.

Thresholding is a statistical operation; such methods neglect all of the spatial information of the image and do not cope well with noise or blurring at boundaries. The most common attribute used is pixel grey-level (intensity), although colour may also be used.

Thresholds may be applied ‘globally’ across the whole image or ‘locally’ within regions of the image. Choosing the appropriate threshold level for a given image can be a problem and a matter of trial-and-error in some cases; indeed the problem of selecting the appropriate parameters is common to many image analysis techniques.

It is rarely possible to identify a single ‘perfect’ threshold level that works equally well across the image. This may be because of changes in lighting quality or focus across the image. When performing a binary threshold operation there are two types of error - ‘false positives’ and ‘false negatives’. Generally a threshold level is chosen that balances these types of error.

Many thresholding techniques are reviewed by Sezgin & Sankur [91]. However, within the scope of this project, thresholding is used as a method by which edges are localised. This is discussed in the following section.

#### **3.4.1 Thresholding as a Means of Labeling Edges**

The techniques of non-maximal hysteresis and edge tracking as applied to the Canny edge detector are a form of thresholding and labelling the edges. The elimination of false edges by thresholding increases the signal-to-noise ratio of the differentiation and smoothing operations and improves the edge localisation. First derivative edge detectors produce an edge response that is a number of pixels across, so some form thinning operation is generally required in order to localise the edge.

The non-maximal suppression algorithm, as used in the Canny detector, provides an improvement to basic skeletonisation methods. The idea is to determine the local maximum of the gradient modulus and then track along the top of the gradient ‘ridge’ in both directions along the orientation of the edge (i.e. orthogonal to the gradient direction).

A pixel is considered if it is the local maximum if the gradient modulus in the direction of the gradient. All edge pixels that are not actually on the top of the edge are set to zero so as to give a line of single pixel width in the output: this process known as ‘non-maximal suppression’.

The tracking process is controlled by two thresholds:  $T_{max}$  and  $T_{min}$  where  $T_{max} > T_{min}$ . Edge tracking can only begin at a point on a locally maximal edge pixel with a gradient modulus higher than  $T_{max}$ . Tracking then continues in both directions out from that point until the height of the ridge falls below  $T_{min}$ .

This edge hysteresis helps to ensure that noisy edges are not broken up into multiple edge fragments. Usually, the upper tracking threshold can be set quite high, and the lower threshold quite low for good results. Setting the lower threshold too high will cause noisy edges to break up. Setting the upper threshold too low increases the number of spurious edge fragments.

### **3.5 Region Detection & Texture Analysis**

Region detection methods are the opposite of edge detection. Such techniques attempt to identify areas that meet some ‘criteria of homogeneity’ – i.e. they look for regions that display similarities in hue, intensity, texture orientation etc. There are two main types of region detection methods: ‘region-growing’[92, 93] and ‘region-splitting’[94].

These methods are brought together in ‘split-and-merge’ detection. Firstly the image is split recursively until each region meets the specified criteria then adjacent regions are merged together if they satisfy the same criteria. In more complex images even more complicated criteria may not be enough to give acceptable results. The split-and-merge techniques introduced by Chen and Pavlidis [95] and later developed by Spann & Wilson [96] use a linked pyramid and statistical decision criteria to combine global and local region information.

Texture information is an important consideration for edge detection. Real world objects often contain some textural component, which presents problems to many edge detection algorithms that may be overcome using region segmentation methods. The detection of texture boundaries may prove to be useful in the context of this project. Combining output from different edge detection algorithms, or the same detector with different scales may give improved results.

Pure texture segmentation gives only a coarse segmentation, so texture segmentation can only be used as an auxiliary tool to check segmentation and texture parameters. Haralick *et al.*[97] provide definitions of texture and derive a number of texture parameters including contrast, correlation, direction, entropy, homogeneity and uniformity.

One approach commonly used to handle texture is to smooth the image using a gaussian or other blurring filter, however this of itself can cause problems as the strength of the blurring filter increases it becomes more difficult to detect the position of the edges accurately (i.e. edge localisation suffers) and also fine detail which should be preserved becomes lost along with the texture ‘noise’.

### **3.6 Image Analysis**

Basic edge detection techniques identify pixels that are potential edges, however they are still only recognised as individual pixels that have no relationship to each other. Once identified, these edge pixels (also referred to as ‘edgels’) must be combined into ‘edge primitives’ in order to provide a more meaningful representation for higher level processing. Various methods of aggregating these edgels have been proposed.

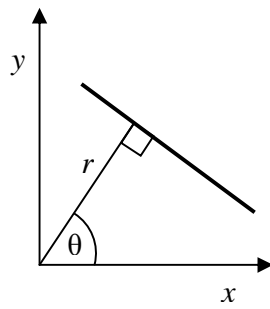
#### **3.6.1 Vectorisation**

Many vectorisation methods have been proposed in the scenario of converting line drawing images (e.g. engineering drawings) directly into CAD models. The output of vectorisation should represent the shape of the original image as faithfully as possible. However, none of the proposed methods work perfectly [98,99]. Many of these methods are based on skeletonisation or other thinning methods that are not required here.

Probably the simplest method of vectorisation is chain encoding (e.g. Freeman Chain Code [100]). Line and arc-fitting algorithms are often employed to convert the original image into a low-level vector format. Line-fitting methods are popular but a ground-truth line cannot be recovered correctly if some parts of it are missing or have serious distortions [101].

#### **3.6.2 Hough Transform**

The Hough transform is one method that is often used to detect / define known geometrical shapes such as lines and circles (or other known shapes [102]) in images. The main advantage of the Hough transform technique is that it is tolerant of gaps in feature boundaries and is relatively unaffected by image noise, however techniques based on the Hough Transform are computationally and memory- intensive [103]. The Hough transform may potentially be used as a means of assigning an ‘edge orientation’ to edge pixels detected using a second derivative detector. The Hough ‘transform space’ is created where points in the ‘Hough space’ map to lines in the image space. The parametric representation of the line is used to construct such a space, as shown in Figure 3-3:



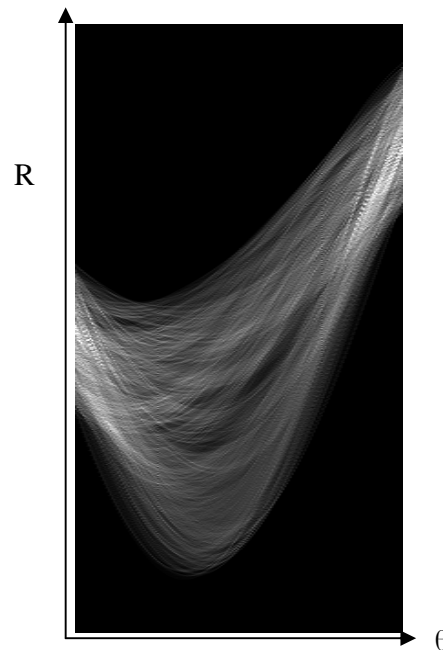
$$x.\cos(\theta)+y.\sin(\theta) = r$$

where  $\theta$  is the angle of the line from the origin orthogonal to the line, and  $r$  is the distance from the origin to the line:

**Figure 3-3a: parametric description of a straight line**

This “Hough space” corresponds to the parameters  $r$  and  $\theta$ , representing lines. Evidence for the presence of straight lines is then accumulated.

This method finds many lines in the image, but has several unwanted effects. Firstly, quantisation of the pixels in the image space and of the accumulator cells in Hough space leads to a ‘bow tie’ effect. A cluster of points in Hough space are mapped back into the image space to form a group of lines of slightly different orientations intersecting at a common point.

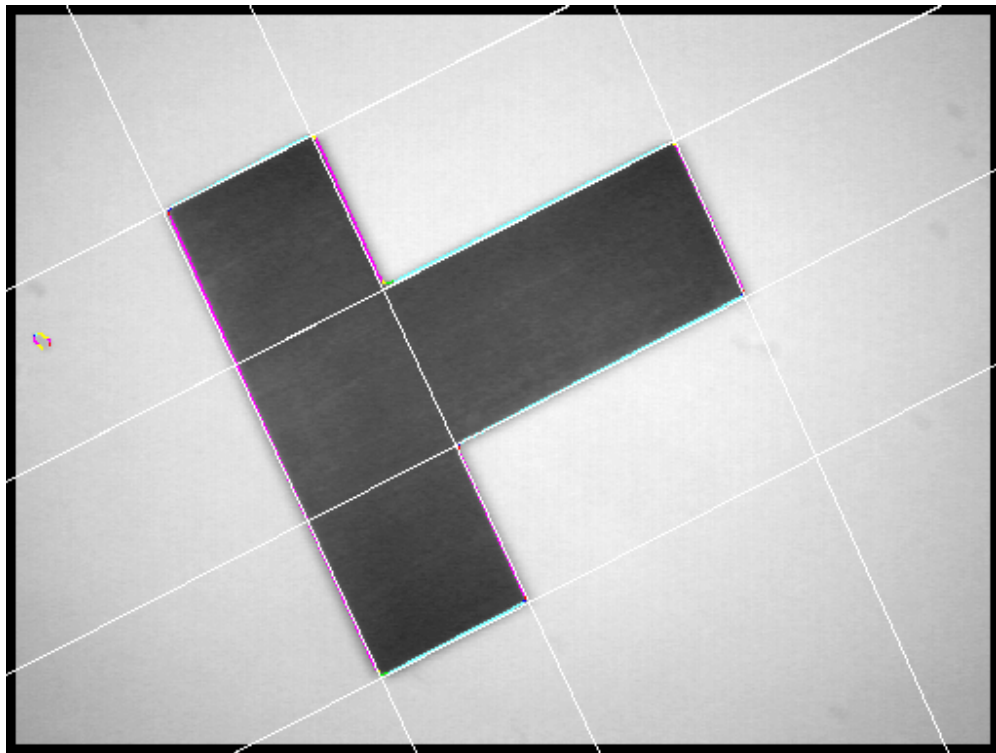


**Figure 3-4: Hough parametric space 2D accumulator array. ( $\theta = 0-180^\circ$ ,  $R$  = distance from origin)**



Secondly, long lines that pass through many pixels are favoured by the accumulator over shorter lines because there is a lot of pixel evidence for them. These shorter lines may be equally valid but simply do not accumulate enough evidence simply because they do not cross enough pixels. It is possible that a long, broken line may accumulate more evidence than a short continuous one and, in noisy images, noise in the central part may actually overshadow valid lines near the boundaries.

Thirdly, the standard Hough transform only gives the parameters ( $r$  and  $\theta$ ) for the line – it does not give endpoints of the line which are required for terminating the line in the image. (See Figure 3-4). A number of authors have proposed improvements to the standard Hough transform to improve line detection and localisation, but many of these are computationally expensive. Ji & Xie [104] review the approaches taken by a number of authors and propose a method by which edge localisation may be improved.



**Figure 3-5: Hough Transform of Canny edge detected lines. The Hough Transform has been thresholded to show only the 'strongest' lines in the image. Note that the Hough lines extend across the whole image.**

### 3.7 Summary

Edge detection is performed by enhancing the image edges with a linear filter that approximates a first or second derivative, followed by a decision step in which a threshold is applied to the edge magnitude image. Prior to the edge detection a smoothing step is often applied in order to reduce the occurrence of ‘noise’ and other minor edges in the image, however the smoothing operation may also cause a distortion of edges and a shift in the reported edge position. The degree of error due to the smoothing operator depends mainly on the ‘strength’ of the operator used (and the shapes of the edges). Some of the ‘classic’ edge detection methods that have been described over the last 20 years were tested for their suitability to this project. These include the Sobel[72], Prewitt [73] and Canny [74] first derivative operators and the Laplacian of Gaussian[77] (LoG) and Difference of Gaussian (DoG) second derivative operators.

Designing a general edge detection algorithm that performs equally well in all contexts and captures the requirements of any subsequent processing stages is still something of a ‘holy grail’. Consequently, a variety of edge detection methods have been proposed that are suited more specifically to the tasks or environments in which they are expected to perform. Edge detectors generally require some operator interaction in the selection of parameters. The Canny detector especially is now widely used across many fields of image analysis and may perhaps be considered as a *de facto* standard. However, the fact that many other edge detector models are still being published suggests that the problem of a universal edge detector that approaches the speed and accuracy of the human eye remains to be solved.

Once edge pixels have been discovered they must be aggregated into meaningful structures, e.g. vectorisation into simple geometric shapes such as straight lines and arc segments. Within this project these vectors are used in order to determine the scan orientation regions. The development of scan regions from these vectors is covered in Chapter 4. The implementation and testing of the edge detection and vectorisation algorithms are discussed in Chapter 5. Edge detection methods are not the only methods available in determining the observed features in an image. In many cases region detection methods may provide a more successful partitioning of the image however due to time constraints these methods were not implemented.

---

57 Ballard, D. & Brown, C. '*Computer Vision*' Prentice Hall Englewood Cliffs, NJ. 1982

58 Gonzales & Woods '*Digital Image Processing* Prentice Hall; 2nd International Ed edition, 2003 ISBN: 978-0130946508

59 <http://www.cs.cmu.edu/~cil/vision.html> (last accessed 23-3-07).

60 Fu, K.S. & Mui, J.K., '*A Survey On Image Segmentation*', Pattern Recognition, vol. 13, pp. 3-16, 1981.

61 Low, A. '*Introductory Computer Vision and Image Processing*' McGraw-Hill, 1991, Chapter 6, p.69.

62 Pitas, I. and Venetsanopoulos, A.N., '*Edge Detector Based on Non-Linear Filters*', IEEE Transactions on Pattern Matching and Machine Intelligence, Vol.8 no.4 pp 538-550, 1986

63 <http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm> (last accessed 10 October 2006)

64 Poggio, T. & Torre, V. '*On Edge Detection*', IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8 no.2, pp.147-163, March 1986.

65 Jain, A.K., '*Fundamentals of Digital Image Processing*', Prentice Hall, 1989.

66 Nalwa, V.S., '*A Guided Tour of Computer Vision*', Addison-Wesley, 1993

67 Zamperoni, P. '*Image Enhancement*', Advances in Imaging and Electron Physics, Academic Press, San Diego, 1995, pp. 1-76

68 Russo, F., '*Edge Detection in Noisy Images Using Fuzzy Reasoning*' IEEE Trans. Instrum. Meas. 47 pp. 1102-1105. (1998)

69 Kundu, A., '*Robust Edge Detection*' Pattern Recognition 23 pp. 423-440

70 Bovik A., Huang, T. & Munson, D. '*Non Parametric Tests for Edge Detection in Noise*', Pattern Recognition, Vol: 19 No: 3, Page(s): 209-219, May 1986

71 Ziou, D & Tabbone, S., '*Edge Detection Techniques - An Overview*', Pattern Recognition and Image Analysis, 8(4) Dec. 1998, pp. 537-559.

72 Lyvers, E.P., and Mitchell, O.R. '*Precision Edge Contrast and Orientation Estimation*', IEEE. Transactions on Pattern Analysis and Machine Intelligence, Vol.10 no.6 1988, pp.927-937.

73 Prewitt, J.M.S. '*Object Enhancement and Extraction*', in Picture Processing and Psychopictories, Eds. B. S. Lipkin and A. Rosenfeld, Academic Press, New York (1970).

74 Canny, J. F., '*A Computational Approach to Edge Detection*', IEEE Trans. Pattern Analysis & Machine Intelligence, Vol. 8, No. 6, 1986. pp. 679-698.

75 Deriche, R., '*Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector*', The International Journal of Computer Vision, Vol.1 no.2 1987, pp. 167-187

76 <http://www.cee.hw.ac.uk/hipr/html/canny.html> (last accessed 10 October 2006)

- 
- 77 Marr, D. and Hildreth, E. C., '*Theory of Edge Detection*', Proc. Roy. Soc. London, 207 pp.187–217 Nov.1980
- 78 Marr, D. '*Vision*' Freeman, 1982, Chapter 2, pp 54-78
- 79 Heath, M., Sarkar, S., Sanocki, T. & Bowyer, K. '*Comparison of Edge Detectors, A Methodology and Initial Study*', Computer Vision and Image Understanding, Vol.69 No.1, Jan 1998. pp 38-54. (Article No. IV960587)
- 80 Shin M., Goldgof, D. & Bowyer, K. '*Comparison of Edge Detector Performance Through Use in an Object Recognition Task*', Computer Vision and Image Understanding, Vol. 84 No.1 pp. 160-178, Oct 2001
- 81 Kanaugo T., Jaisimha, M., Palmer J. & Haralick, R. '*A Methodology for Quantitative Performance Evaluation of Edge Detection Algorithms*', IEEE Trans. Image Processing, Vol. 4 no.12, pp. 1667-1673, Dec 1995.
- 82 M. Heath , S. Sarkar, T., Sanocki, & K. Bowyer, '*A Robust Visual Method for Assessing the Relative Performance of Edge Detection Algorithms*', IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.19, no.12, pp 1338-1359, Dec 1997
- 83 Freeman, H. & Davis, L. S. '*A Corner Finding Algorithm for Chain Code Curves*', IEEE Transactions on Computers, 26, pp.297–303 (1977).
- 84 Beaudet, P. R. '*Rotational Invariant Image Operators*', Int. Conference on Pattern Recognition, pp.579–583. 1978.
- 85 Moravec, H. P., '*Visual Mapping by a Robot Rover*', Joint Conference on Artificial Intelligence, pp.598–600, 1979.
- 86 Harris, C. G. & Stephens, M. '*A Combined Corner and Edge Detector*', 4th Alvey Vision Conference, pp.147–151 1988.
- 87 Zheng, Z., Wang, H. & Teoh, E. '*Analysis of Gray Level Corner Detection*', Pattern Recognition Letters, 20, pp.149 162 (1999).
- 88 Kitchen, L. & Rosenfeld, A. '*Gray-level Corner Detection*', Pattern Recognition Letters, 1, pp.95–102 (1982).
- 89 Wang, S.J. & Binford, T.O. '*Generic, Model-based Estimation and Detection of Discontinuities in Image Surfaces*'. In ARPA IUW, volume II, pp.113-116, 1994.
- 90 Smith, S. M. & Brady, J. M., '*SUSAN - a new approach to low level image processing*', International Journal of Computer Vision, 23(1), pp.45–78 (1997)
- 91 Sezgin, M. & Sankur, B., '*Survey Over Image Thresholding and Quantitative Performance Evaluation*', Journal of Electronic Imaging Vol 13. no.1 January 2004, pp.146-165
- <sup>92</sup> Adams, R. & Bischof, L., '*Seeded Region Growing*', IEEE Trans. on Pattern Analysis and Machine Intelligence, vol 16, no. 6, pp. 641-647, June 1994
- 93 Z. Lin, J. Jin, and H. Talbot, '*Unseeded region growing for 3d image segmentation,*' in ACM International Conference Proceeding Series, Selected papers from the Pan-Sydney workshop on Visualisation, vol. 2, 2000, pp. 31–37.

- 
- <sup>94</sup> Ohlander, R., Price, K., & Reddy, D., '*Picture Segmentation Using a Recursive Region Splitting Method*', Computer Graphics and Image Processing, volume 8, 1978 pp. 313-333
- 95 Chen, P. C. & Pavlidis, T. '*Image Segmentation as an Estimation Problem*'. In Image Modelling, pp.9–28. Ed. A. Rosenfeld, Academic Press, 1981.
- 96 Spann, M. & Wilson, R. G. '*A Quad-Tree Approach to Image Segmentation Which Combines Statistical and Spatial Information*'. Pattern Recognition, 18(3/4):257–269, 1985.
- <sup>97</sup> Haralick, R. Shanmugam, K. & Dinstein I., '*Texture Features for Image Classification*'. IEEE Transactions on Systems, Man, and Cybernetics, SMC- 3(6): pp.610-621, 1973
- 98 Tombre, K. '*Analysis of engineering drawing: state of the art and challenges*' Graphics Recognition - Algorithms and Systems. K. Tombre, and A.K. Chhabra (eds)., Springer-Verlag, Berlin, Germany, 1998. , pp. 257-264.
- 99 Tombre, K. & Tabbone, S. '*Vectorization in Graphics Recognition: To Thin or Not To Thin*'. Proceedings of ICPR'00. Vol.2, pp. 91-96, 2000.
- 100 Freeman, H., '*Boundary encoding and processing*,' in Picture Processing and Psychopictories. New York: Academic, 1970.
- 101 Hori O. & Tanigawa. S. '*Raster-to Vector Conversion by Line Fitting Based on Contours and Skeletons*' Proceedings of ICDAR'93, pp. 353-358, Tsukuba, Japan, 1993.
- 102 Ballard, D. H., '*Generalizing the Hough Transform to Detect Arbitrary Shapes*' Pattern Recognition, Vol.13, No.2, 1981. pp. 111-122
- 103 Risse, T. '*Hough Transform for Line Recognition: Complexity of Evidence Accumulation and Cluster Detection*.' CVGIP, 46(3): pp. 327-345, 1989.
- 104 Ji, Q. & Xie, Y. '*Randomised Hough Transform with Error Propagation for Line and Circle Detection*', Pattern Anal Applic., Vol 6. pp. 55-64, 2003

## **4. Improved Scanning Using Data Fusion Based On Image Analysis**

The edges determined by the image analysis methods described in Chapter 3 are positions where geometric edges are potentially present. In this chapter it will be shown how the output from the various image analysis methods will be used to develop the scan orientation path over the object.

Distortions may also be caused by sudden variations in reflectivity and this method is expected to reduce these distortions as well as either rapid changes in either gradient or reflectivity will usually cause an edge in the camera image. Path planning is the key aspect to the successful outcome of this project as it will determine the orientation of the laser relative to the object in order to avoid occlusion and secondary reflection errors in the range image.

This chapter describes the outline for the work in this thesis and provides a conceptual overview of the scan region algorithms, the hardware required to perform the scanning process and also covers calibration and registration of the system in order that the camera and laser images may be successfully correlated.

#### **4.1 Development of Conceptual 'Data Fusion' Solution**

The basic concept of this work is to provide a means by which the laser scanner has some knowledge of where problem areas may exist within the scan object, and thereby take some action to avoid the occurrence of errors in those areas. The output from the various image analysis methods employed will be used to develop the scan orientation path over the object.

From previous work (see Section 2.2) it has been shown that, if the laser sensor can be kept parallel to geometric edges in the object, then the occurrence and magnitude of scan distortions are minimised. This may be achieved by rotating the laser scan head during the scan process. Previous work in the department required that the whole object was scanned multiple times. (In his thesis, Wong showed that occlusion and reflection errors could be minimised by combining scans from a number of different orientations and determining which sets of data contained errors at a certain location by comparing all the scans. Wong used 8 complete scans of the object at 45° separations in order to identify the best orientation which was effective but extremely time-consuming). The method for this project will remove the requirement for much of the redundant repetition of scans.

The basic idea is to construct scan regions around edges detected in the image, using the orientation of the detected edge to provide guidance for the orientation of the laser. Initially the parts of the object around those edges parallel to the sensor are scanned, then the laser is then rotated and another partial scan is completed, until the whole object has been scanned.

It has been shown that image processing methods can be used to provide a means of detecting edges in a photographic image (see Chapter 3) and that these detected edges can be converted into vectors with a known orientation relative to the axes of the image.

This section shows how these vectors may be developed into scan regions suitable for the region of interest to be scanned at the locally optimal orientation using the single perspective laser sensor, thereby minimising the scan errors.

#### 4.1.1 The Concept of ‘Orientation Alignment Tolerance’

Ideally the object should be scanned whilst adjusting the laser to keep it precisely parallel to the edge for each scan point along the edge. However, this is often unnecessary and impractical (and in some cases impossible,<sup>i</sup>) as the laser does not need to be absolutely parallel to the edge in order to avoid errors. As long as the laser orientation remains within a few degrees either way, then the occlusion and reflection effects are minimised (as shown in Chapter 2, Section 2.4.2.3). By grouping edges at similar angles (within the orientation tolerance) into a single scan it is possible to reduce the number of partial scans required to cover the whole object without introducing significant scan artefacts, thus reducing the total number of required scans.

Orientation (or angle) alignment tolerance is a measure of how far, in terms of difference in angle, the sensor may be rotated away from parallel alignment to an edge before significant errors are discernable in the scan data.

What may be considered a ‘significant’ error depends somewhat on the scale of the object and the eventual purpose for performing the scan. The error tolerance for a precision machine part may be much smaller than that of (for example) a bathroom tile.

Considering the diagram in Figure 4-1 overleaf, the colours in the circle represent the likely ‘goodness’ of the resulting data for the laser orientation, with green indicating a good orientation and red indicating a likely occurrence of errors with respect to the edge ‘A’.

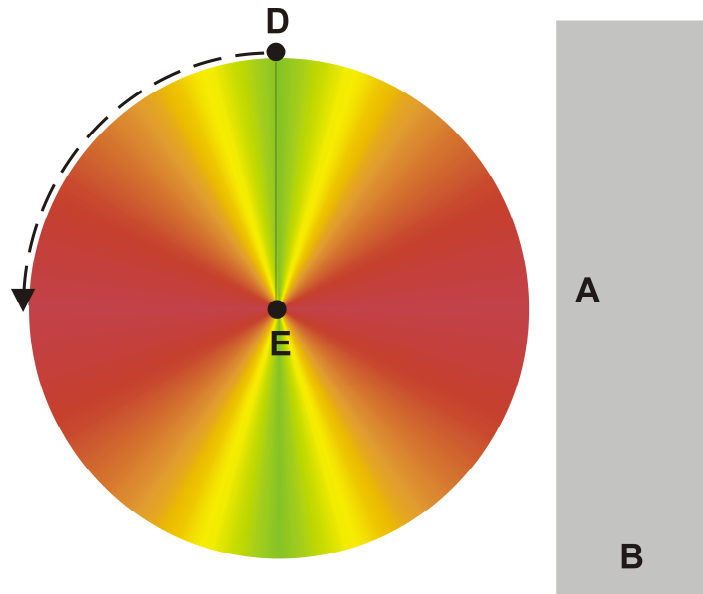
As the orientation of the laser detector ‘D’ is rotated about the emitter position ‘E’, away from parallel to the edge the magnitude of distortions increases rapidly until distortions are at their worst with the laser perpendicular to the edge. For the edge ‘B’ the situation is shifted by 90 degrees.

---

<sup>i</sup> The laser is mounted on a stepper motor and the minimum rotation step angle limits the precision with which the laser can be aligned to an edge. Also where two or more edges at different orientations meet it is not possible to keep the laser parallel to both edges, but a ‘compromise’ angle may suffice to avoid significant errors from either edge.



Within the green region (with respect to edge A) the magnitude of the errors may be considered ‘acceptable’ for the application, although the determination of the orientation alignment tolerance arc for any given object is non-trivial as there are a number of contributing factors that interact to affect the data output by the laser sensor.



**Figure 4-1: A ‘plan view’ of laser orientation tolerance with respect to the edge ‘A’ of the grey block. ‘E’ represents the position of the emitter, ‘D’ represents the position of the detector. The green area is the extent of the ‘good’ orientation alignment tolerance with respect to edge A**

These factors include the profile (in cross section) and height of the edge, the height of the sensor above the object, relative reflectivity characteristics (e.g. colours and textures) of the surfaces involved, as well as sensor characteristics (e.g. the distance of the sensor to the emitter, the angle of view of the detector optics and the width of the detector element).

The decision to extend the allowed orientation tolerance may not be entirely based on scan accuracy, as the time to complete the scan set may also be a factor in commercial systems: fewer orientation groups means fewer scans are required to complete the whole set at the potential expense of some scan errors being present. In the test platform, the user may choose the angle orientation tolerance for the current object to examine how the composite output appears for different levels of tolerance.

#### **4.1.2. Method 1 - Dominant Scan Orientation**

The simplest case is to perform a single scan at an orientation determined to be ‘optimal’, thereby minimising the number of potential errors (with respect to a single scan). This ‘dominant orientation’ maximises the number of edges to which the laser is parallel, or rather, to maximise the total edge length at that orientation during a single scan.

Whilst minimising the time taken (as only a single pass over the object is required) this will still result in a significant number of errors for most objects, since there will almost always be a number of edges at orientations for which the chosen scan orientation is not optimal. However the total number of errors for the chosen orientation should be less than for any other scan orientation. This method is also used as a baseline to which the relative improvement of other methods can be compared.

The method simply determines the total edge length by orientation, with the number of potential orientations decided either by the minimum rotation step of the sensor head, or by the ‘angle tolerance’ of the object. The edge lengths may be summed either by counting pixels by orientation or by determining the length of each vector. It has the advantage that no calibration information is required – the required processing occurs entirely within the image domain.

There is a level of user-input involved in the decision for the object angle tolerance that may alter the output to some extent. By limiting the number of orientations that are considered it is possible to influence the decision of which orientation is dominant. For example, with an object where the majority of the edges fall within two or three major orientations, but it is unclear which of the orientations is dominant, limiting the number of orientations will force any smaller contributing edges at orientations close to (but not matching) the major orientations into being considered as contributing to these major orientations. Errors for these edges may occur but are likely to be smaller than for the ‘competing’ orientations.

### **4.1.3 Method 2 Algorithm – Multiple Partial Scans**

The method described in the previous section is a simplistic solution that does not make very good use of the information available from the image analysis. A much better approach is to perform a number of partial scans, chosen so as to minimise distortions.

Instead of performing a single scan across the whole of the object area, the information from the vectors can be used to perform a set of partial scans at different orientations, with each scan covering just the parts of the object where the errors will be minimised at that particular orientation. The set of partial scans can then be integrated to provide a complete point cloud representation of the object with much reduced errors.

#### **4.1.3.1 Partial Scans Based on Scan Regions**

As the first ‘real’ scenario, it was hypothesised that the image edge vectors could be used as templates defining the axis of ‘scan regions’ corresponding to the position of the ‘real’ edges in the point cloud.

The primary axis and orientation of the region is provided by the edge vector and the region is extended to either side (perpendicular to the vector) by a number of pixels calculated to include the region in which the largest extent of bow waves that may occur. The spike effect is more ‘localised’ to the edge so will also be covered by this region.

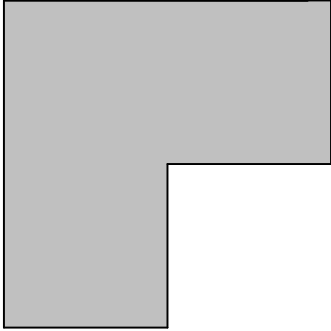
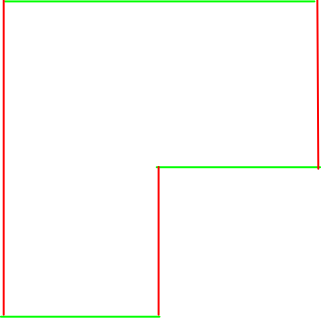
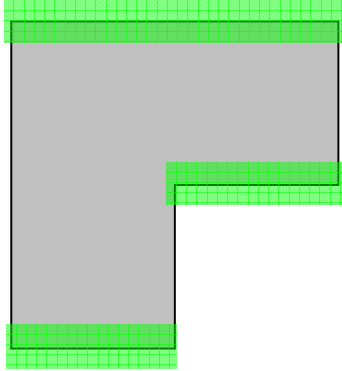
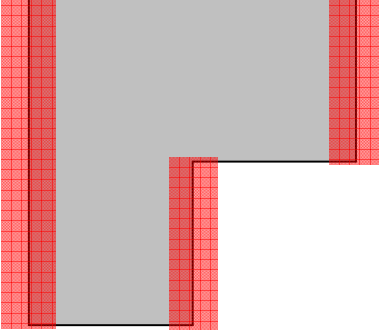
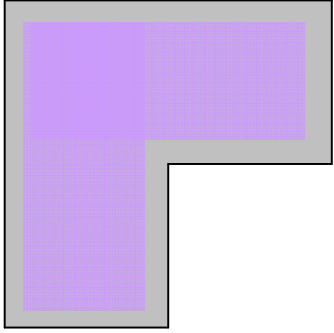
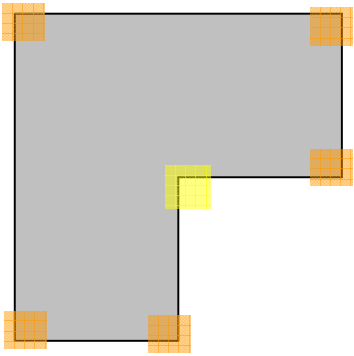
From an output perspective, for each orientation, the active scan regions provide a template for a partial scan of the object; with the sum of the scan orientations covering the whole object. For areas where there are no orientations with an active scan region the dominant scan orientation (from Method 1) should be used as the ‘default’ scan orientation in order to ensure there is at least one value for each point in the scan space.

The partial scans are then mapped over the default scan, replacing the default values where the orientation scan layers provide ‘better’ orientation data, resulting in a composite output scan image. To further improve efficiency it would be possible (in some cases) to subtract the areas covered by the other scan orientation regions from the default direction scan. It may also be possible to plan for scan regions where there are no detected features to be scanned at a lower scan resolution, which would contribute to decreasing the total scan time.

Figure 4-2 shows how the scan path planning may be resolved for a simple L-shaped object.

- a) Represents an image of the object to be scanned. This will be edge-detected using one of the methods discussed in Chapter 3.
- b) The edge-detected image is vectorised. These vectors are then expanded into scan regions.
- c) Represents the areas required to be scanned with the laser parallel to the x-axis in order to avoid occlusion and reflection errors
- d) Represents the areas required to be scanned with the laser parallel to the y-axis in order to avoid occlusion and reflection errors
- e) Represents the area where scanner orientation is not critical according to the available edge information
- f) Highlights areas where there are likely to be overlapping scan region values which will require post-scan resolution. Those areas of scan region overlap indicated in orange areas are expected to be resolved satisfactorily. However, the yellow (internal corner) is likely to be more difficult to resolve as it will suffer from problems due to secondary 'bow wave' reflections from the vertical faces of the object in both scan orientations. However this corner is approached by the laser there will always be secondary reflections.

N.B. in general the software cannot determine if a corner is 'internal' or 'external' by the edge detection process alone. Some method of pre-scan profiling may help to determine such characteristics.

	
<p>a) Representation of 'L' shaped object. This image is passed through an edge detection filter.</p>	<p>b) Edge vectors are determined. Red lines represent edges detected at 0 degrees, green lines represent edges detected at 90 degrees</p>
	
<p>c) The 'green' vectors are expanded into 'scan regions' around each 90 degree vector. Region width is dependent on the object height relative to the scanner</p>	<p>d) Red vectors are expanded into scan regions where orientation must be aligned with Y-axis in order to avoid occlusion errors along these edges.</p>
	
<p>e) Lilac area indicates where scan head orientation is non-critical. For this area scanning at any orientation will not cause errors according to available edge information</p>	<p>f) Areas indicated in orange show where overlaps must be resolved. The yellow area indicates an 'internal corner' where no orientation will avoid all potential errors.</p>

**Figure 4-2 (a-f) scan region orientations applied to a simple 'L' shaped object**

### **Determination of Scan Region Width**

When an edge is detected there is no knowledge of which side of the edge may have distortions present. The height of the edge influences the width of the potential region of error (as described in Chapter 2) and to be sure of eliminating all the errors the laser should be kept parallel to the edge for the whole width of the scan region on the lower side of the edge. The edge represents a change in height (or reflectivity) but image analysis can not determine the relative heights of the surfaces on either side of the edge from a single camera image.

Errors on the ‘high’ side of the edge tend to be well-localised to the edge, whereas errors on the ‘low’ side of the edge have the potential to extend for some distance from the edge (especially for secondary reflection errors), but without prior knowledge of the object’s topology it is impossible to know which side of the edge is ‘higher’ or ‘lower’ so the scan region must be extended by an equal amount on both sides. More detail on the determination of the region width is provided in Section 6.2.2.2

### **Resolution of Scan Points with More Than One Active Region**

Many positions in the scan area will be represented by just one active scan region, however in other positions there will be multiple active orientations where regions overlap. Because the scan regions are ‘grown’ around edges some positions in the overall image may ‘belong’ to more than one scan region; this will be the case particularly where edges at different orientations meet. Determining the single output value from these different orientations requires some decision about which values are most likely to be ‘correct’.

The simplest method is just to take a (mean) average of the values for each orientation to provide a single output value. If there are more than two orientation values available it may be possible to use a weighted average or discard outlying values, although it is possible that the outlying value may be the one that is ‘most correct’. (This may occur when there are a number of edges in close proximity and overlapping scan regions of a similar orientation ‘outvote’ a single scan region of a different orientation even though the considered position is closer to the outvoted region vector than either of the others).

For any case where multiple values exist for a single point it is likely that the output value will be a compromise between the values in the input set rather than the ‘single best’ value as there is no way to determine which value is ‘most correct’.

Furthermore, such areas of overlap also increase the total required scan area (i.e. the overlaps increase the overall scan time as the same area must be revisited for each orientation). Depending on the number of overlapping regions and their required scan width, this may add a significant amount of time to the overall scan. These considerations led to the development of the scan algorithm described in the following section.

#### **4.1.3.2 Partial Scans Based on Nearest Vector Orientation**

For many positions in the scan base a clear choice exists as to which orientation will give the best value because it lies in only one scan region. However, where scan regions overlap, the above method requires that some compromise output is required.

The method described in this section provides some determination of which value is more likely to be correct by selecting the orientation for each scan point based on the orientation of the vector to which it is closest. Combining this method with the previous method for areas of scan region overlap may also be possible.

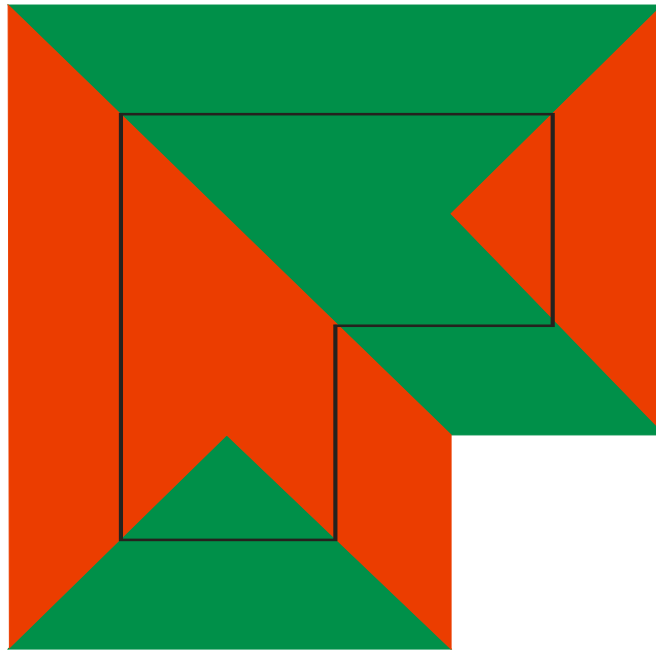
Given the nature of the scan distortions that are to be avoided it seems reasonable to assume that, for any point in scan space, the output value will be most affected by the edge that is in closest proximity. As it is not possible to determine the relative object surface heights or reflectivity from the camera image it must be assumed at this stage that all edges are equal in their effect on scan output.

Determining the orientation for each pixel on the basis of vector proximity is likely to be much slower than the previous method due to the time taken to compare each scan point to all the vectors (within a maximum radius of effect from the current scan point). However this method obviates the requirement to average the results of a number of different orientations, although methods are also suggested here by which a weighted average value may be determined if required

- 1) Keep top ‘N’ closest vectors (regardless of orientation band).
- 2) Keep nearest vector from each orientation band within region of influence.

In these cases, the output value is decided on by weighting the each contributing value proportionately according to the distance from the vector(s) at that orientation.

It is likely that, in a commercial system, some degree of blending will be required between the partial scans in order to disguise where the regions join. This has not been done in this experimental system because any errors that the process fails to remove should not be disguised.



**Figure 4-3 Partial Scan Regions by Nearest Vector for L shaped block (indicated by black line). Using this strategy there are no overlapping regions – each position in the scan is assigned an orientation based on the orientation of the nearest vector.**



#### **4.1.4 Method 3 Single Scan with Rotating Scan Head**

In an ideal system only a single scan would be required, with the laser sensor being continually rotated to the optimal angle for the current scan position. This idea is sound in principle, however practical requirements mean that in practice it is very hard to achieve, because of the speed of the scanning head relative to the speed of laser rotation.

With the current system architecture it is not possible to slow down the movement of the scanning head in order to allow time to rotate the sensor to the correct orientation for any given point, because the laser is dependent on moving at a constant speed in order to sample the object at regular intervals over the surface.

This may be acceptable where the laser is moving from a region where the orientation is non-critical if it can be ‘pre-oriented’ in time before reaching the orientation-critical region, but if it is moving from one ‘orientation-critical’ region to another then there will be a period where the orientation is incorrect for both regions as it rotates from one orientation to the other.

If the scan speed and rotational velocity of the laser head are known and constant then it would be possible to calculate the distance required to rotate the scan head from one orientation to another. This ‘transitional’ region could then be rescanned at the critical orientations involved and the corrected region values patched into the overall scan. However, this negates the ‘single scan’ advantage of this method and there are other issues involving laser calibration such as the rotation ‘pirouette’ (considered in Section 4.3) that make this method more difficult to implement using the hardware that is currently available for this project.

If the laser alignment could be more precisely controlled and the sample rate of the laser sensor could be varied as the speed of the sensor head over the object is varied then this method would be viable, but given the limitations of the available equipment and existing time constraints it was decided that this method would not be implemented.

## **4.2 Hardware Requirements**

This section describes the hardware available for this project including the requirements for image capture and the rotation of the laser scanning head. One of the primary considerations for this project is the ‘point cost solution’ aspect; that is, the solution should take advantage of the advent of low-cost digital imaging technology to provide an improvement in the scan quality. The development of an integrated solution should not add a significant financial overhead to the overall system.

### **4.2.1 Current Hardware**

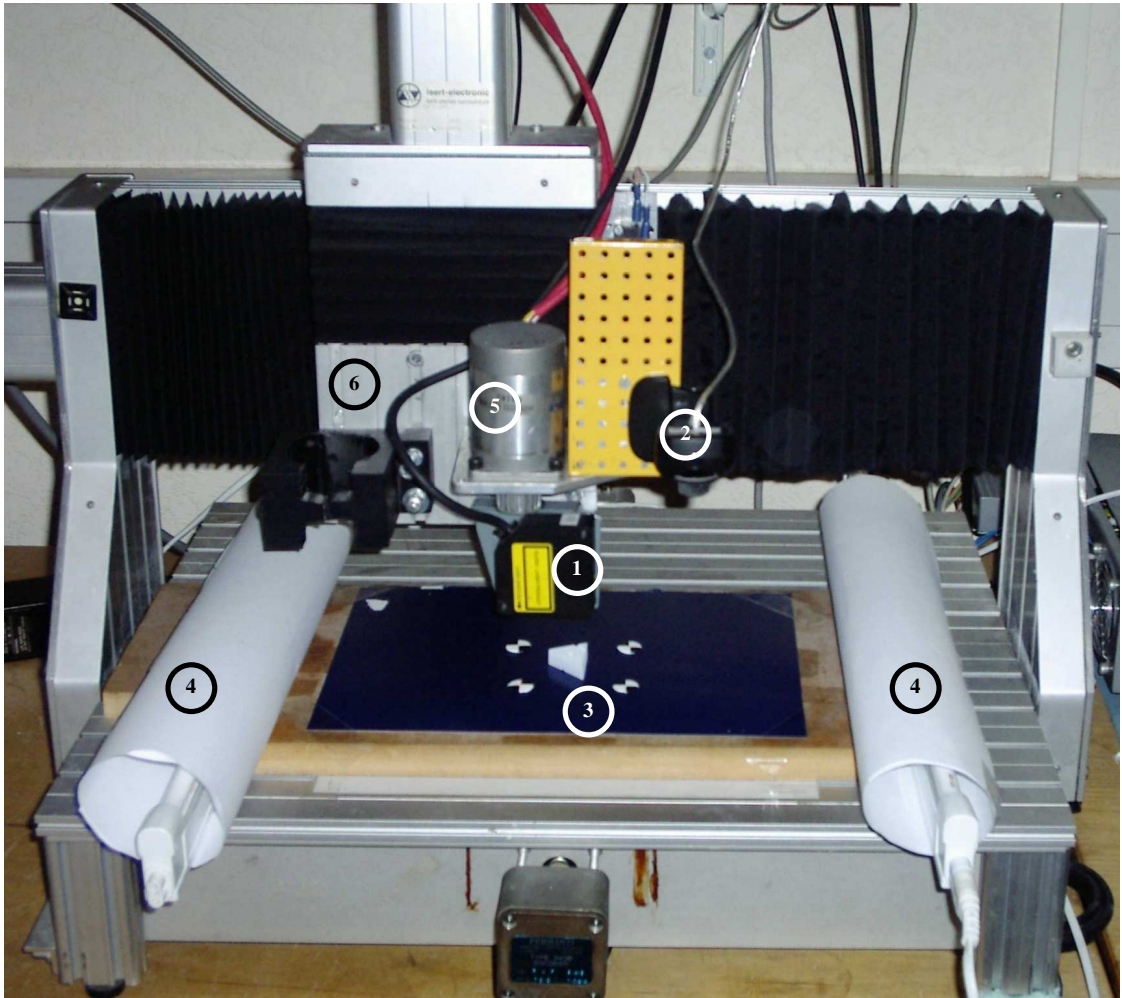
The current hardware setup comprises of a CNC platform with a laser sensor mounted on a moving gantry. The movement of the gantry is controlled via a controller card from a Windows NT-based PC running Axiomatic’s Scan3D software. Sensor data is sampled one line at a time and stored temporarily in the laser controller card before being transferred to the Scan3D process. Scan3D is also used to define basic machine calibration settings, and provides storage and interpretation of the scan data and conversion of the scan data into a number of file formats.

#### **4.2.1.1 Isel CNC Platform**

The main components of the current system comprise of an Isel CNC machine with 3 axes of movement: the gantry is able to move in the ‘x’ and ‘y’ directions to any point in the scan bed and the ‘z’ direction raises and lowers the gantry mount to the correct height above the bed. Soft sponge ‘feet’ were used to dampen mechanical vibration, in order to reduce the stochastic (noise) distortion in the range image.

The experimental rig was situated in a low lighting environment, to reduce the possibility of stray light interfering with the readings by the detector. Ideally the detector should only detect the spot image projected by the emitter onto a surface with high diffuse reflectivity.

Figure 4-4 shows the main components of the scanning hardware.



**Figure 4-4: Image showing the major components of the test rig.**

- 1) Laser Sensor
- 2) Camera on mounting plate
- 3) Sample object on scan bed. (Calibration markers are placed around the object)
- 4) Fluorescent Light Tubes with diffuser 'lamp shades'
- 5) Stepper motor controlling axis of rotation for laser sensor
- 6) Moving gantry of CNC machine

Axiomatic Technology (UK) Ltd's Scan3D software was used to manage the scanning procedure, from calibrating the rig to recording and saving the captured range images as data files. The movement of the CNC machine is controlled by an Axiomatic Technology AxController USB Card which receives instructions from the Scan3D software via a controller task on a standard PC running Windows NT.

Note that the Axiomatic controller card is a relatively new design developed during the evolution of this project. The original controller used a standard serial bus and was much slower in operation, especially with respect to data transfer rates. However the introduction of the new card has revealed timing issues in the hardware that have caused some problems in data integrity during this project (see Section 4.4).

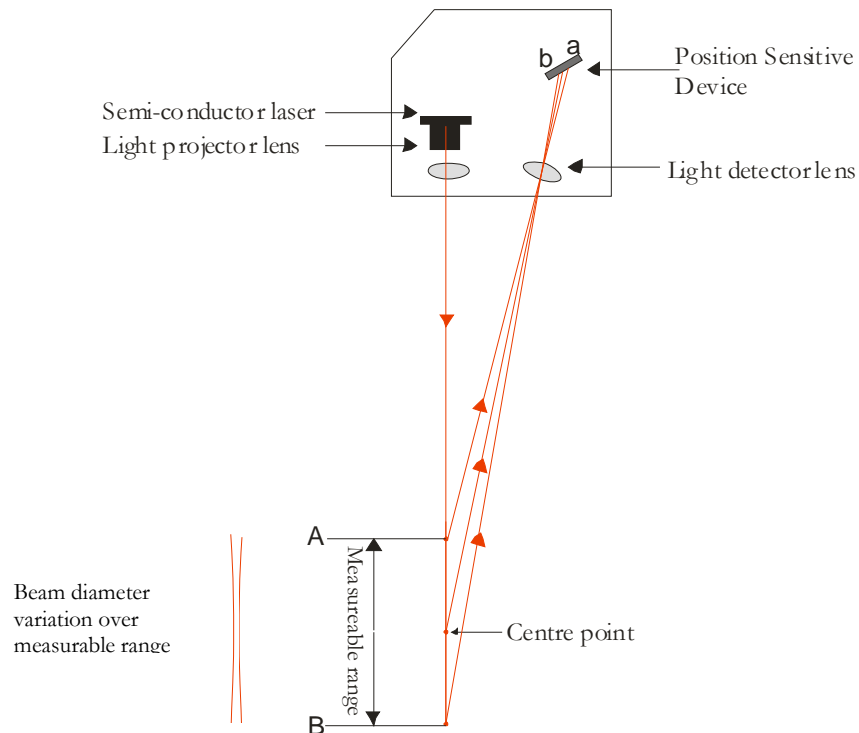
#### **4.2.1.2 Matsushita Laser Scanner**

The laser scanner used in this project is a Matsushita ANR1151 single perspective laser triangulation scanner with a standoff distance of 50mm and a range of +/-10mm. The laser diode operates at 685nm wavelength and maximum resolution of the scanner is 5 $\mu$ m. At the standoff distance the triangulation angle  $\theta$  is 20°.

This is a (relatively) cheap single-point laser sensor that exhibits the problems for which this project aims to provide a solution. Other laser scanners are available which use multiple detectors (e.g. Wolf & Beck) and / or multiple emitters to compensate for occlusion errors, however these systems are correspondingly more expensive than the Matsushita laser and do not solve the problem entirely.

The diode emits a beam of light that is focused by the projector lens onto the object of interest. A proportion of the light reflected from the target object is focused by the detector lens and casts a light spot on the position sensing device (PSD). The position of the spot varies depending on the displacement of the target object. An analogue voltage is output, corresponding to the displacement of the target within the measurable range. By measuring the fluctuations in the position of the light spot, the distance to the target object may be calculated.

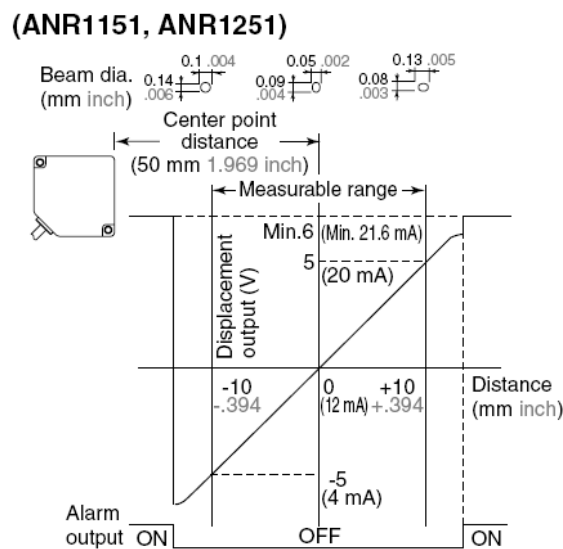
The size of the spot is dictated by the optical design, and influences the overall system design by setting a target feature size detection limit. The limitation imposed by the beam diameter on the minimum feature size that can be detected. For example, if the spot diameter is  $50\mu\text{m}$  it will be difficult to resolve a lateral feature of less than  $50\mu\text{m}$  in dimension. Spot diameter is specified at the centre and extremes of the measurable range. Limitations of physical optics dictate that the beam waist varies throughout the working range. The collimating lenses focus the beam to a minimum waist at the standoff distance; at the extremes of the measurable range the beam diameter is larger. The same rule of feature size detection applies at the extremes of the measurable range, as the beam waist is larger, so the smallest detectable feature size is larger.



**Figure 4-5: Diagrammatic representation of the laser sensor, showing the main components.**

The standoff distance is measured from the projector lens to the centre point, with the measurable range evenly distributed about that point. The beam width (spot diameter) is minimal at the centre point and increases towards the extremes of the measurable range.

The PSD is a single-axis analogue detector that converts reflected light into continuous position data [106,107]. PSD-based systems are efficient and provide the high data rates. The processing required to perform the triangulation is simple. They also have rapid gain control – an important consideration when dealing with surfaces of varying texture, colour and reflectivity.

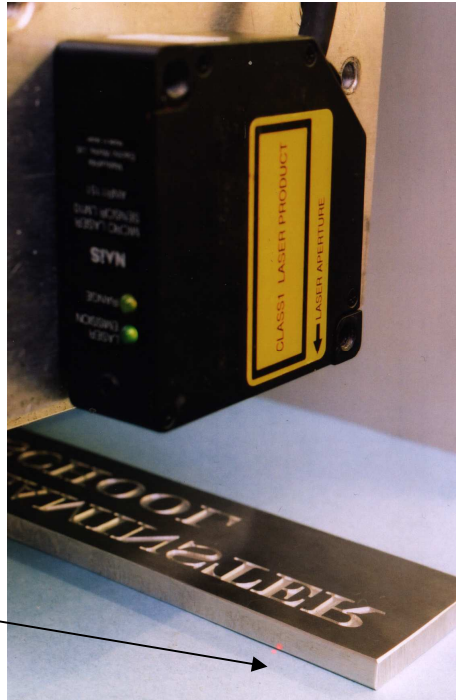


**Figure 4-6: Diagram showing the fluctuation in voltage representing the change in measured distance over the range of the laser detector [108].**

According to Wong [1], one disadvantage of PSDs is that the detector determines the centroid of the spot image. If two spots are present the detector will report a single ‘weighted’ position of both spots. Another drawback is that PSD systems are very sensitive to spot intensity. This is inherent in the detector and can be accommodated by additional circuitry. The effect of this sensitivity is that if the spot intensity changes while the spot position remains the same, the calculated position of the spot may change. This is typical where a change in surface reflectivity occurs.

Stray reflections, where the projected spot image on a highly reflective surface, may result in a small amount of stray light reflected to the detector. The combination of such stray reflections with the primary spot image causes the detector to produce erroneous signal values[109].

Position of  
laser spot



**Figure 4-7: Matsushita NAIS ANR1182 sensor in use, scanning an inscribed metal block.**

## **4.2.2 New Hardware Requirements**

In order to undertake the required elements of this project the current system must be extended to provide a means by which the laser head may be rotated to the optimal angle and to capture an optical image of the object that will be scanned in order to determine the presence and location of the edges.

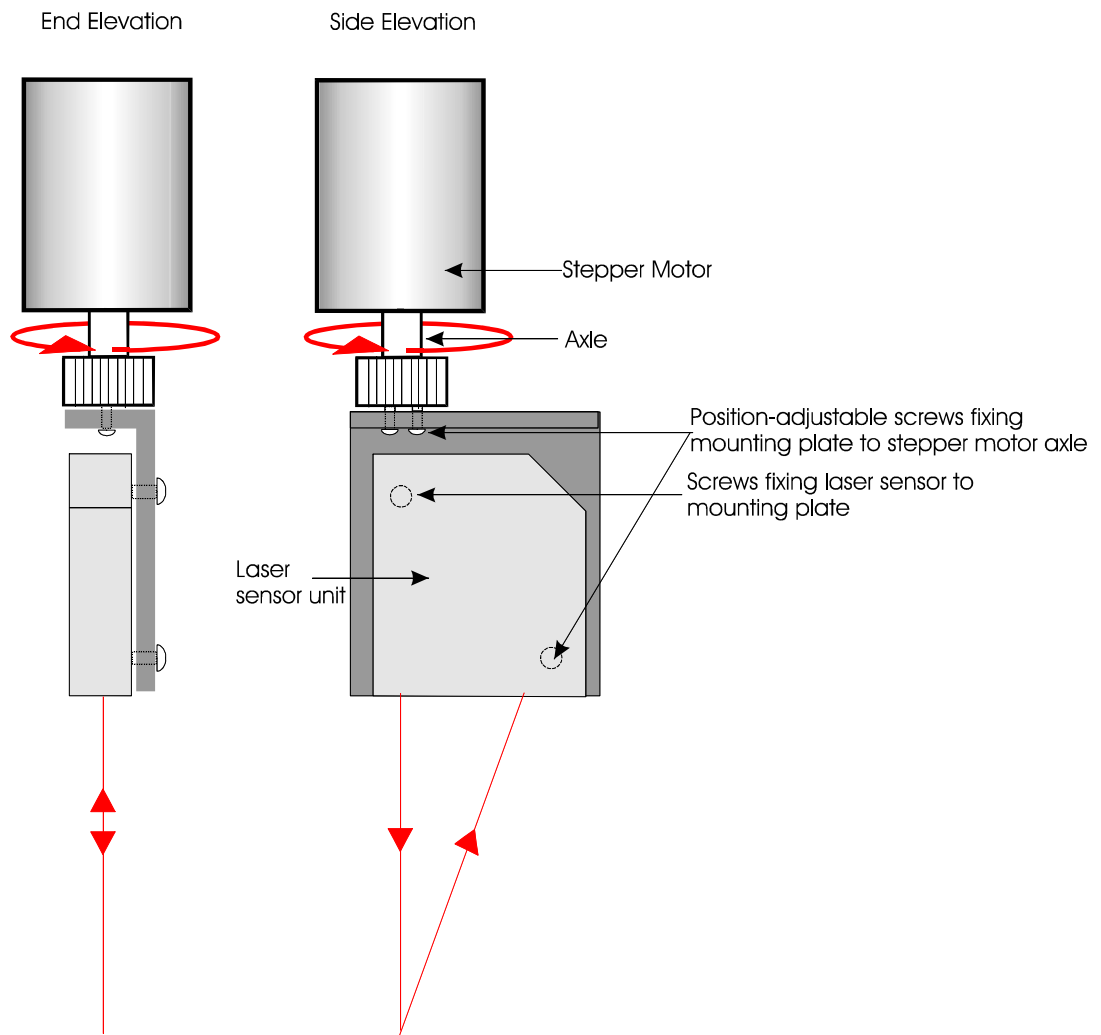
### **4.2.2.1 Fourth Axis of Movement for Laser Scanner**

In order for the laser head to be rotated to the optimal orientation for the partial scans, a fourth axis of control was integrated to the CNC machine, comprising of a stepper motor mounted on the gantry block. The laser sensor is mounted onto the stepper motor, allowing the rotation of the sensor. The stepper motor used in this project allows 200 discrete steps per 360° rotation (i.e. one ‘step’ rotates the laser by 1.8°).

The laser sensor is fixed on the stepper motor through a mounting plate, the position of which may be adjusted in order to minimise displacement and skew of the laser as it is rotated. Unfortunately, the automatic rotation of this motor was not realised during the development time of the project, however it was still possible to test the effect of rotating the scan head manually, simply by turning the scan head to the required orientation by the required number of steps.

Care must be taken when mounting the laser on to the stepper motor axle to ensure that the axis of rotation for the scanner is co-axial to the laser projector, and that the emitted beam is perpendicular to the scan bed and remains so throughout its rotation. If the laser does not conform to these conditions then errors in the calibration may lead to problems in combining partial scan sections. These issues are discussed in greater depth in Section 4.3





**Figure 4-8: Diagram showing two views of the laser sensor mounting on stepper motor axle. This assembly is mounted on the z-axis gantry of the CNC machine, allowing the height of the laser to be controlled precisely.**

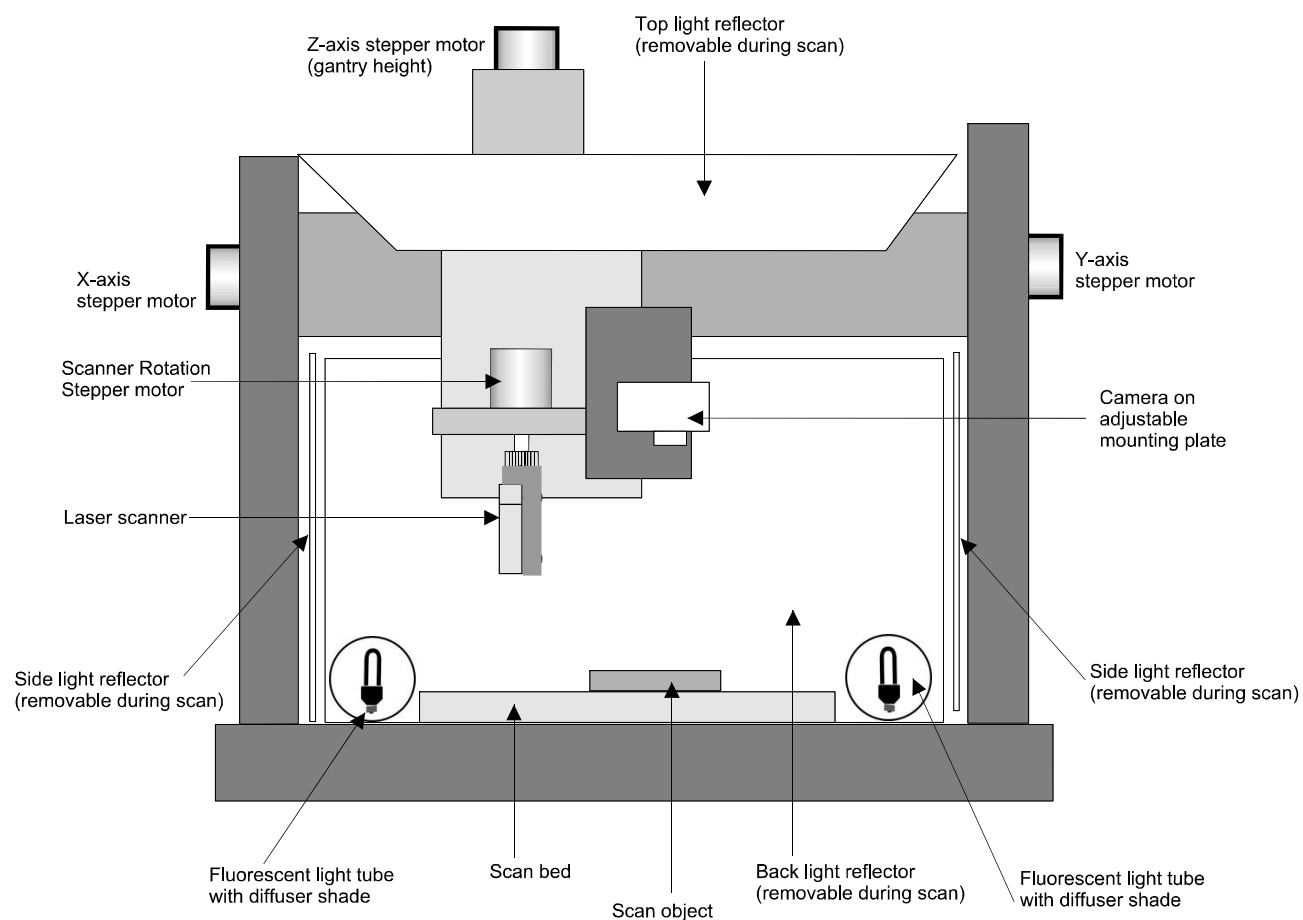
#### **4.2.2.2 Lighting**

Early tests showed that control of the lighting in the scene was very important in providing useful images for the edge detection process. The results from ambient lighting conditions were poor, from the aspect of scene contrast and the occurrence of shadows.

There are somewhat conflicting requirements in lighting the object under consideration. Harsh, directional lighting that will cast shadows in the image must be avoided, as these will also be detected as edges in the image analysis process (and therefore scan regions will be developed that have no bearing on the original object location), however the scene must be strongly and evenly illuminated to provide sufficient contrast that the ‘true’ object edges may be easily discerned. A completely diffuse light source would make it very difficult to detect geometric edges. What is required is a soft, multidirectional source of illumination.

It was decided to use a pair of 20W fluorescent light tubes 30cm long. These tubes were then wrapped in a single layer of 80g/m<sup>2</sup> A3 paper to provide a more diffuse light source.

These were placed equidistantly on either side of the object, close to the edges of the scan bed. To further reduce the directionality of the light sources, large white reflector cards were placed around the CNC machine to create a ‘light box’ around the object. The lights and reflectors can be removed once the camera image has been captured so they do not obstruct the scanning process (or cause issues with stray light in the laser detector).



**Figure 4-9: Schematic of CNC machine showing location of lights and reflector boards.**

#### **4.2.2.3 Optical Imaging Device**

In order to acquire an image from which edge orientation data may be extracted, the camera must be mounted co-axially with the laser on the scanning platform. The camera used to capture the images from which the image analysis process determines the location of edges in the scene is one of the key components in determining the success of the project.

The height of the camera over the scan bed can be adjusted to make best use of the image area for objects of different sizes. One of the criteria for this research is the development of a 'point-cost' solution (i.e. that the cost of the laser and camera and software should together be significantly less than that of the more expensive lasers with internal data averaging). During the period of this research the quality of available digital image capture technology has significantly improved and the cost of such devices has fallen dramatically.

A number of cameras have been investigated during the development of the system. Initially a Pulnix monochrome CCD video camera owned by Axiomatic was used for the image capture process, attached to a Data Translation frame-grabber card in the PC (with an image resolution of 768 x 576 pixels). This provided a reasonable quality monochrome (greyscale) image, however the camera was not very good under the available lighting conditions due to the relatively small maximum aperture of the lens. The camera had only a manual focus and no interface to adjust the image brightness or contrast digitally. Additionally it was rather heavy and bulky by current standards, which required a more robust mounting to be fitted to the CNC gantry.

A PC webcam was tested as an alternative image capture device. This is a cheap USB device for the home PC market. It has a maximum resolution of 640x480 pixels in a 24-bit RGB colour format. The camera is supplied with a software control interface for setting contrast and brightness, and a 'continual capture' image that allowed for easy positioning of the camera over the object

However it also only has a manual focus and shows a poor depth of field at the close range used here (i.e. under 0.5m). Another drawback in its consideration for this task is the slight 'wide-angle' focal length, causing a small degree of 'fish eye' effect to be present in the captured images. A number of test images also showed that the quality of the optics in the

camera were quite poor, resulting in slight ghosting of the image, probably due to some internal reflection.

As the limitations and quality issues of the webcam became apparent, a Concord 4060AF digital camera was also used in the latter stages of testing. This provided the opportunity for a much higher resolution image (up to 4 million pixels) if required (although the increased image size would also increase image processing overheads). The camera optics proved far superior to those in the webcam and the video camera, especially at close range where the macro setting proved useful. Whilst the digital camera was not immune to lens distortions the errors were limited to the periphery of the image and were less severe than those of the webcam.

The drawback of this camera is there is no direct interface to the PC whilst capturing the image (i.e. no 'real-time' image available on the PC), although there is a image window on the back of the camera, it is more difficult to align the camera over the object whilst not being able to see the camera position at the same time. Also, the images have to be downloaded from the camera to the PC as a separate task, rather than being captured directly to the computer.

Although colour imaging was available for the webcam and digital camera, the first camera used was only capable of greyscale ('black and white') imaging and hence the early software edge-detection development was limited to greyscale only. Due to time constraints and the complexity of colour edge detection this was not updated during the project. Colour edge detection remains a possible line of investigation that is mentioned in the Future Work chapter.

The cameras used in this project represent 2 examples at the bottom end of the range of available devices. Any TWAIN or USB compliant imaging device (allowing simple programmatic control of the image capture process) could be used.

### 4.3 Calibration & Registration Issues

Image registration is the process of transforming the different sets of data into a single coordinate system and is a necessary requirement in this system in order to be able to correlate the data obtained from the camera imaging system with the laser scan data.

Calibration and registration of the laser and camera image are of great importance in this project. Even if the camera image provides a ‘perfect’ map of where the edges of the object appear, if the information from the image cannot be referenced to the laser scan coordinates with sufficient accuracy then the system will fail to select the correct laser orientation for some positions.

Furthermore, a number of issues were identified relating to the rotation position of the laser sensor head that may cause further problems, both with the reported values and the challenge of matching a set of partial scans.

In image registration the coordinate system of the original image is often referred to as the *reference image* and the image to be mapped onto the reference image is referred to as the *target image*. For the purposes of this project the laser scan space is considered to be the reference image and the camera image space is the target image: i.e. the scan bed coordinates are considered ‘absolute’ and the image coordinates are transformed to match those of the scan<sup>ii</sup>.

#### 4.3.1 Laser Calibration

As described in Section 4.2.2.1, the laser is mounted on a stepper motor, allowing the sensor to be rotated in order to change the orientation. This freedom of movement and the mounting of the sensor on the stepper motor introduce the potential for misalignments in the system. Incorrect alignment of the laser will cause the measured points to deviate from the ‘true’ coordinates (with reference to the CCM gantry position). Although the effects caused by these misalignments may be individually small, they become noticeable when multiple scans of differing orientations are combined, especially if the borders of the scan regions meet on a geometric edge.

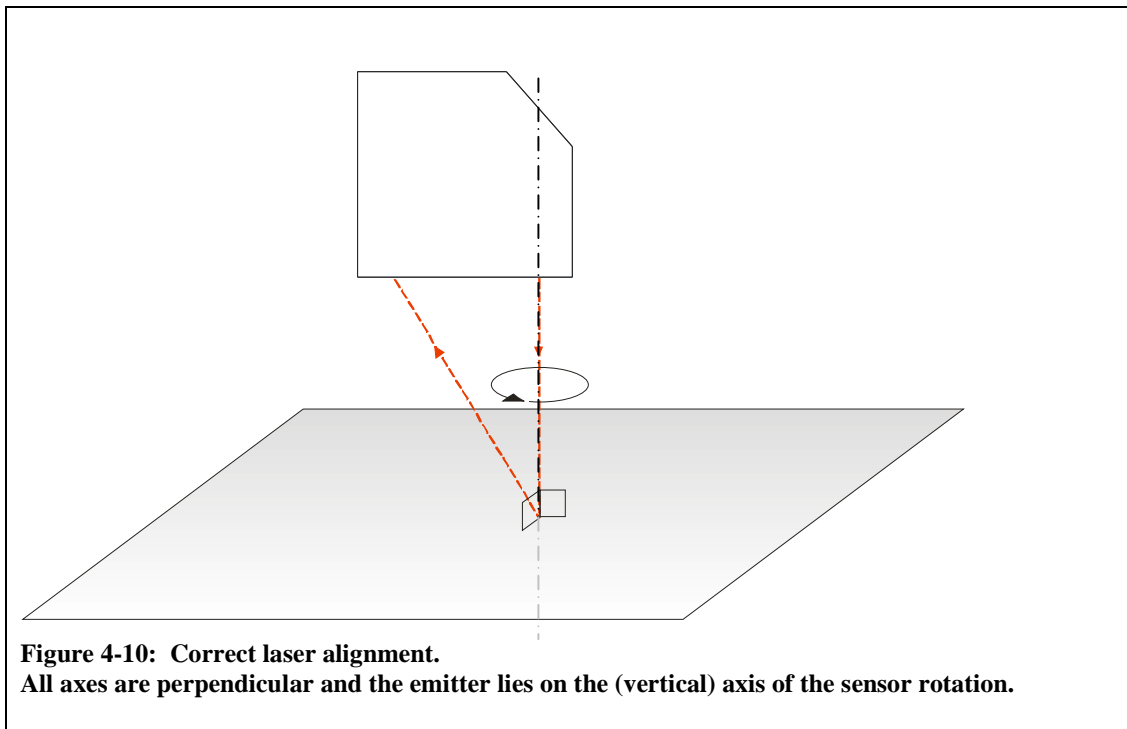
---

<sup>ii</sup> The scan file is the ultimate output file, and so it is easier to consider transformations between the coordinate systems from this perspective when working with the files, both in terms of scan resolution and coverage (i.e. the image must cover the whole of the scan area but there will usually be some peripheral area of the image that may not map to any valid position within the scan).

There are 3 main problems associated with mounting the laser head on a rotating platform: these problems are described in the following sub-sections, where the first scenario describes the ideal case.

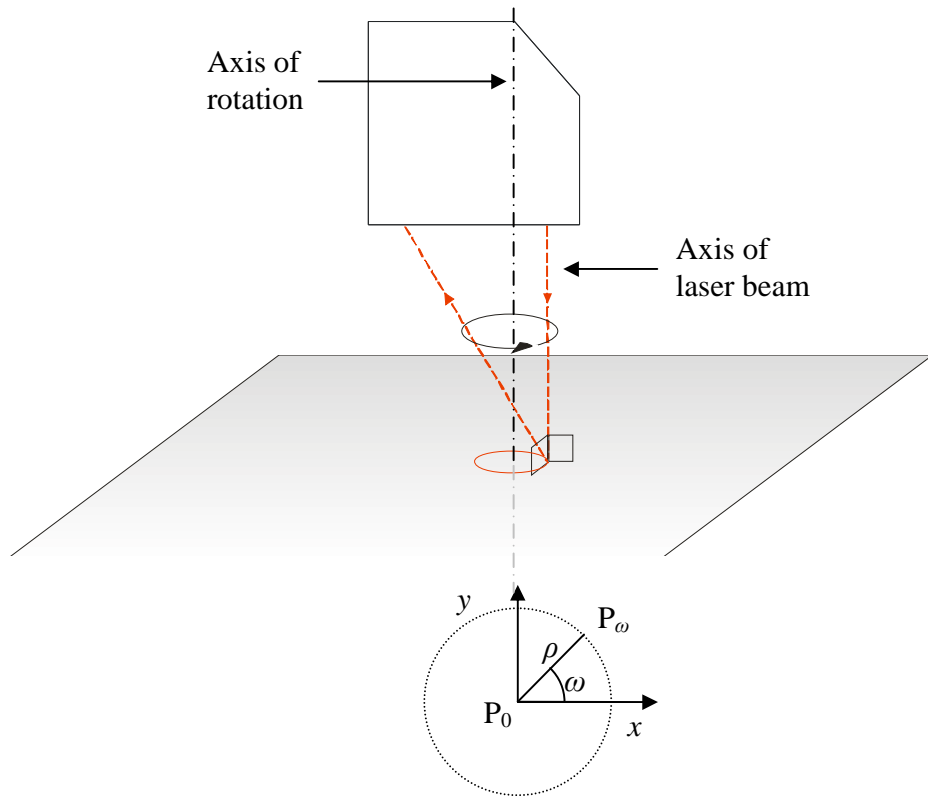
#### 4.3.1.1 ‘Ideal’ Correct Laser Rotation

In Figure 4-10 all the components are oriented correctly. The laser is rotated around the axis of the emitter and both the laser and the axis of rotation are perpendicular to the plane of the scan. In this ideal situation the triangulation measures the correct height at the intended x,y coordinates and no corrections are necessary.



#### 4.3.1.2 'Off Centre' Rotation of the Laser

The simplest error case considered is when the axis of the laser emitter is off-centre with respect to the stepper motor axis of rotation (see Figure 4-11). In this situation the laser spot will describe a circle around the axis of rotation. The z-value (object height) is measured correctly but there is an offset error for any measured height between the true CMM x,y coordinates and the actual measured point. In this example all the other alignments are correct.



**Figure 4-11: Laser rotates around the axis of rotation with offset  $\rho$**

Let  $x_m$  and  $y_m$  be the measured position of x,y coordinates of the emitter, and  $z_m$  is the measurement obtained (relative to the base) then for any position of the sensor x,y at orientation  $\omega$ , and with offset distance  $\rho$ , the actual position of the laser spot,  $x^*, y^*$  is calculated by

$$x^* = x_m + \rho (\cos \omega)$$

$$y^* = y_m + \rho (\sin \omega)$$

$$z^* = z_m$$



The radius of rotation  $\rho$  can be measured by aligning the laser spot on a reference point then rotating the laser through a known angle,  $\omega$ . The laser spot position should then be adjusted so that it again lies on the reference point to give the offset in  $x$  and  $y$ .

#### **4.3.1.3 Laser Beam Not Perpendicular to Scan Bed**

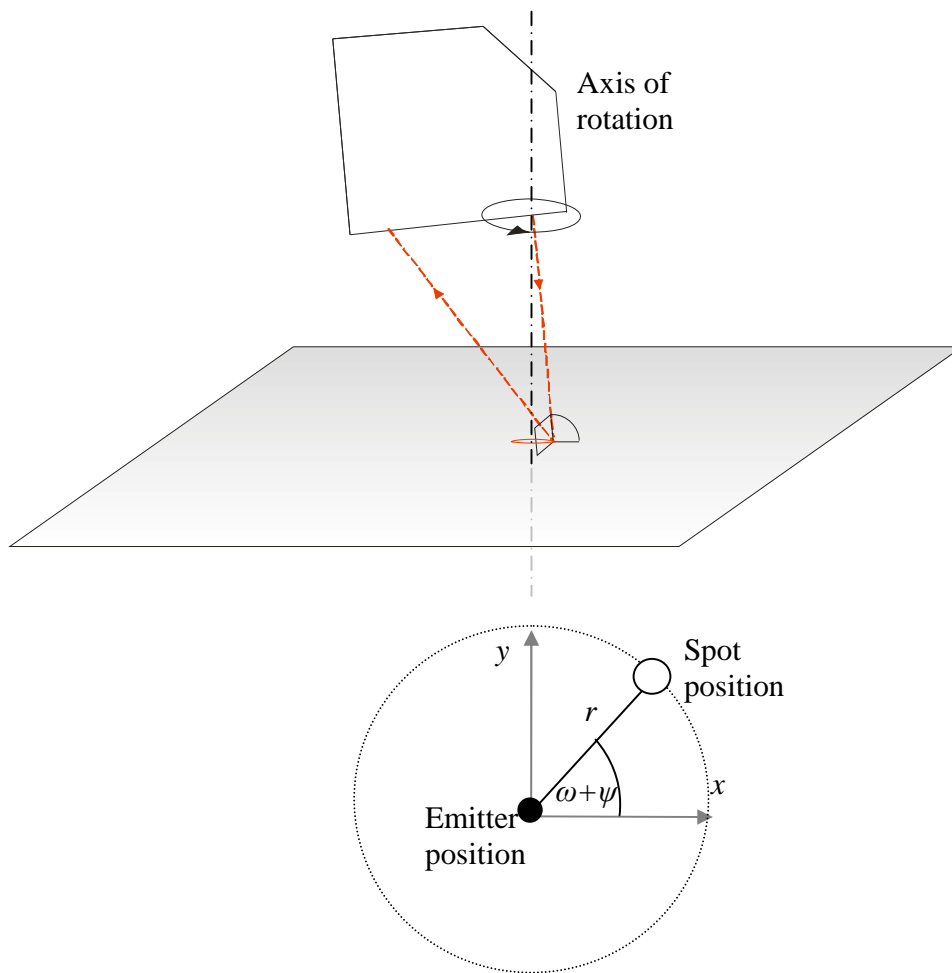
In this situation the laser scanner is tilted, resulting in a beam that is not perpendicular to the base. This results in a small translation of the measured point from the true position. However the axis of rotation is still vertical, so as the laser is rotated, the measured position describes a circle around the axis of rotation. The magnitude of the error between the intended (CMM) coordinates and the actual measured position will vary with the height measurement.

This situation can be considered in two parts: in the simpler case, as shown in Figure 4-12, the axis of rotation passes through the emitter. In this case the spot describes a circle around the axis of rotation with an offset proportional to the angle of tilt.

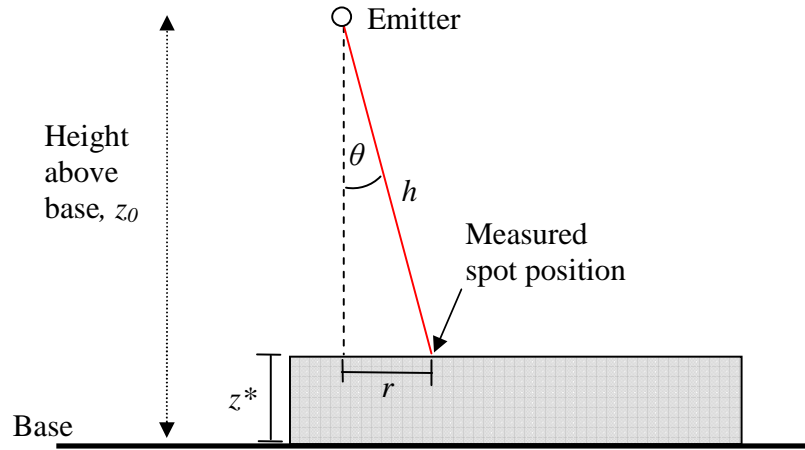
The measured height will be slightly greater than the true height, because the measured distance is also the hypotenuse of the triangle (as shown in Figure 4-13).

The offset distance,  $r$  can be used to correct the height reading if the angle and direction of tilt can be measured. This may be done by observing the magnitude of the change in position of the laser spot in the  $x,y$  axes with a known change in height.

As the measured height changes there will also be a change in the translation of the  $x,y$  measured point relative to the true position (i.e. the magnitude of  $r$  varies with measured height if  $\theta$  is non-zero).



**Figure 4-12 Laser tilted relative to the plane of the scan bed. In this scenario the emitter is on the axis of rotation but the tilt of the laser means that the spot describes a circle around the axis.**



**Figure 4-13: Laser sensor tilt affects the measured height and the measured position also varies with measured height**

Let  $x_m$  and  $y_m$  be the measured position of the  $x,y$  coordinates of the emitter. Let  $z_m$  be the measurement obtained (relative to the base) and  $z_0$  be the height of the emitter above the scan base.

There are 3 components to consider with respect to the correct  $x,y$  position: the direction and the angle of the tilt of the laser (relative to the  $x$  and  $y$  axes) and the rotation of the laser,  $\omega$ .

When  $\omega$  is zero, let the tilt of the beam be in the vertical plane which intersects the  $x,y$  plane at an orientation  $\psi$  relative to the  $x$  axis, then for the general orientation  $\omega$ , the tilt is in the vertical plane which intersects the  $x,y$  plane at an orientation  $\omega + \psi$  relative to the  $x$  axis, and then the offset distance  $r$  can be found by,

$$r = h \sin \theta$$

where  $h = z_0 - z_m$  (i.e. the actual distance of the spot from the emitter)

Then the corrected position  $x^*, y^*, z^*$  can be calculated as follows:

$$x^* = x_m + h \sin \theta \cos (\omega + \psi)$$

$$y^* = y_m + h \sin \theta \sin (\omega + \psi)$$

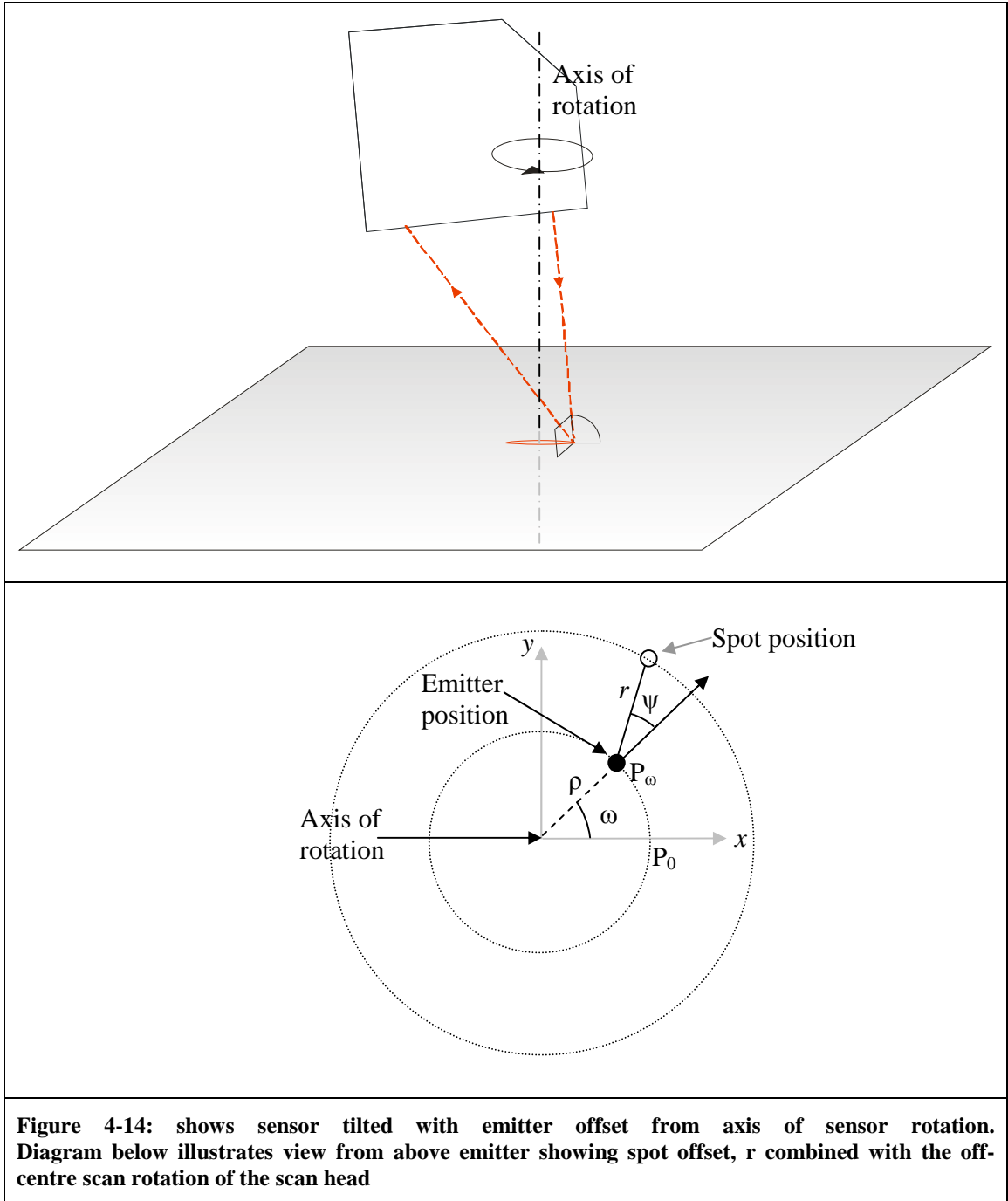
$$z^* = z_0 - h \cos \theta$$

A laser tilt error as small as one degree will result in a positional discrepancy of approximately 1mm and a height discrepancy of 0.01mm at the stand-off distance for the sensor used (i.e. 50mm). The height discrepancy may be negligible compared to other error components in this system, however the positional discrepancy is a concern unless corrected: even at a coarse scanning resolution of 0.5mm this represents an offset of 2 scan lines.

In the more general case the emitter does not lie on the axis of rotation. In this case the emitter rotates about the axis with radius  $\varrho$ . Allowing for a rotation of the scan head where the axis of rotation is not about the emitter, (as described in Section 4.3.1.2) then the combined scenario exists, as shown as shown in Figure 4-14.

As mentioned above, the tilt angle,  $\theta$ , and the direction of tilt,  $\psi$ , can be measured by observing the change in position of the laser spot with respect to the x and y axes as the laser height changes (without rotating the scanner). Once the tilt is known then  $\varrho$  can be found as in Section 4.3.1.2. (In fact, the situation described in 4.3.1.2 can be considered as a specific case of this scenario where the axes rotation and the emitter are parallel).

There may be one sensor height (within the range of the laser) for which the beam intersects with the true axis of rotation and for that height the measured x,y position of the spot will be correct, as the described circle will have zero radius. (i.e. only in the case where  $r = \varrho$  and  $\psi = 180^\circ$ ). However, for all other sensor heights there will be a displacement error. Again, the laser tilt error will result in a positional discrepancy and so any tilt should be kept as small as possible through careful setup of the equipment.



Referring to Figure 4-14, as the scan head is rotated, relative to the x-axis, through an angle  $\omega$ , the angle of tilt,  $\psi$  relative to the scan head is constant, so the direction of tilt is  $\omega + \psi$  relative to the x axis.

As the scan head is rotated by an angle  $\omega$  with the radius of rotation  $\rho$ , then the corrected position for  $x, y$  can now be calculated as follows:

$$h = z_0 - z_m$$

$$x^* = x_m + h \sin \theta \cos (\omega + \psi) + \rho(\cos \omega)$$

$$y^* = y_m + h \sin \theta \sin (\omega + \psi) + \rho(\sin \omega)$$

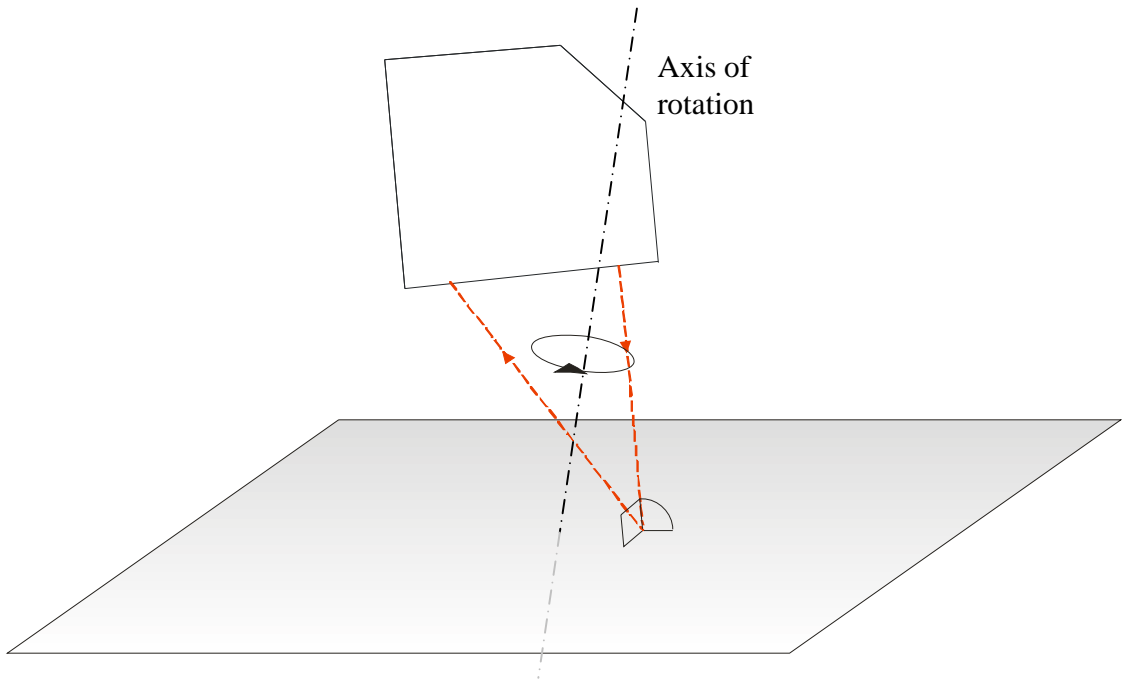
$$z^* = z_0 - h \cos \theta$$

This is true as long as the axis of rotation is perpendicular to the scan bed. The following section describes the scenario when both the laser beam and the axis of scanner rotation display some degree of deviation from perpendicular.

#### 4.3.1.4 Laser Beam and Axis of Rotation Not Perpendicular to Scan Bed

This last scenario is the general case where both the laser beam and the axis of rotation are not perpendicular to the plane of the scan bed. In this case, if the tilt of the laser scanner is zero relative to its axis of rotation (i.e. the beam is parallel to the axis) then the measured point will describe an ellipse on the plane of the scan bed. (i.e. plane of the scan bed intersects the laser as a conic section).

In the previous case, the axis of rotation was perpendicular to the scan bed and the point where the beam hits the object has a consistent angle offset  $\psi$  in the x,y plane. However, if the laser is tilted whilst the axis of rotation is not perpendicular to the plane of the scan bed, the angle offset in the x,y plane changes as the sensor is rotated (although it is constant in the plane perpendicular to the axis of rotation)



**Figure 4-15 illustrates the general situation where the axis of rotation and the laser sensor are both oriented at some arbitrary (non-coaxial) angle to the scan bed.**

The determination of the corrected scan position for this scenario is somewhat more complicated than the previous examples, as now the height of the laser relative to the plane of the scan bed varies as the sensor is rotated. The equations to calculate the corrected position of the laser spot are given overleaf. The full derivation of the algorithm for this scenario [110] is provided in Appendix A.

If x, y and z are the machine axes and  $x_m, y_m, z_m$  are the measured coordinate values, then first the following terms are defined.

$z'$  is the axis of rotation of the scanner and  $\omega$  is the angle of scanner rotation (about  $z'$ ) relative to a zero position in line with the x-axis.

$\theta_s$  is the angle of tilt of  $z'$  relative to z.

Let  $x'$  be the axis perpendicular to  $z'$  in the plane of tilt then, relative to  $x', y', z'$

$\varphi_s$  = angular position of  $x'$  (relative to the direction corresponding to  $\omega=0$ )

$\theta$  = angle of tilt of beam relative to  $z'$

$\psi$  = angle of direction of beam tilt from  $z'$  axis (in  $x'y'$  plane relative to  $x'$ )

$\rho$  = radius of rotation of emitter

$z_o$  = height of centre of rotation of emitter,

then the following terms can be defined:

$$C = \rho \cos (\omega - \varphi_s)$$

$$S = \rho \sin (\omega - \varphi_s)$$

$$h = (z_o - z_m)$$

$$d = h \cos \theta + \tan \theta_s (C + h \sin \theta \cos (\omega - \varphi_s + \psi))$$

$$r = d \tan \theta$$

$$C' = r \cos (\omega - \varphi_s + \psi)$$

$$S' = r \sin (\omega - \varphi_s + \psi)$$

$$\lambda = (h \cos \theta) / d$$

$$X = (C + \lambda C') \cos \theta_s - \lambda d \sin \theta_s$$

$$Y = S + \lambda S'$$

$$\varphi_s^* = \tan^{-1} (\tan \varphi_s \cos \theta_s) \quad \text{this is the angle in the x,y plane corresponding to } \varphi_s$$



then, rotating back by  $\varphi_s^*$ , the corrected positions may be expressed as:

$$x^* = x_m + X \cos \varphi_s^* - Y \sin \varphi_s^*$$

$$y^* = y_m + X \sin \varphi_s^* + Y \cos \varphi_s^*$$

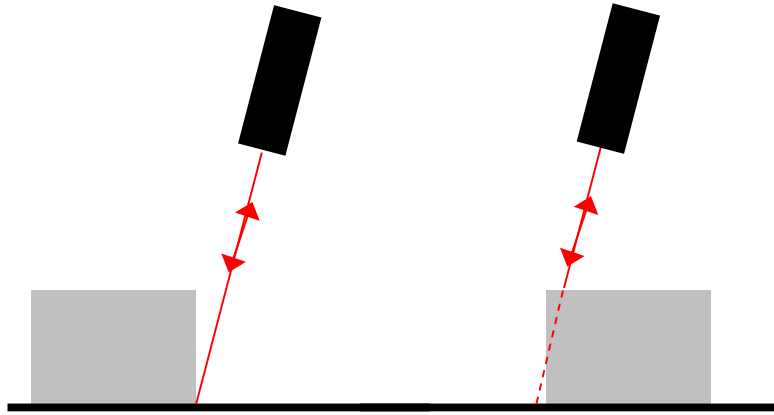
$$z^* = z_o - d \cos \theta_s$$

$$\omega^* = \varphi_s^* + \tan^{-1}(\tan(\omega - \varphi_s) \cos \theta_s)$$

if  $\theta_s = 0$  then  $\lambda = 1$  and  $d = h \cos \theta$ ,  $r = h \sin \theta$ ,  $\varphi_s^* = \varphi_s$  and  $\omega^* = \omega$

it can be shown that the same result as in Section 4.3.1.3 is achieved\*\*

It is important that  $\theta_s$  and  $\theta$  are kept as small as possible. Some measurement errors (due to occlusion or reflection) will be increased as the plane of the beam and sensor is further from vertical and in this case aligning the sensor parallel to the edges will not help to avoid occlusion, as shown in Figure 4-16.



**Figure 4-16:** illustrates an ‘end-on’ view of the scanner aligned with the edge of an object, but tilted (through an exaggerated angle). In (a) the tilt angle does not cause any problems, assuming that the positional corrections are applied. However in (b) the tilt causes the beam to be occluded by the tilt of the sensor

### 4.3.2 Image Registration and Scaling

Recovering the 3D structure from the camera image requires that the camera be calibrated. A camera is said to be calibrated if the mapping between image coordinates and directions relative to the camera centre are known. However, the position of the camera in space (i.e. its translation and rotation with respect to the scan coordinate system) is not necessarily known. Considerable work has been done in both photogrammetry and computer vision to calibrate cameras lenses for both their intrinsic perspective parameters and distortion patterns. Some successful methods have been proposed by Tsai [111] and Faugeras [112,113].

The primary consideration for this project is the correlation of the camera image to the CMM scan point cloud. The calibration aspects of the camera, such as radial distortions (barrel or pincushion distortions) were considered but not implemented here as they were not thought to be a significant factor. In a commercial system the coefficients for the camera calibration would need to be determined and applied to all of the images prior to any analysis. (This would apply to the calibration images as well as the images used to determine object geometry)

A conversion method is required between the image and scan coordinate systems. The image typically represents a small area of the overall scan bed and may fully encompass the scan object, or cover just a part of it, depending on the size of the scan object and the field of view of the camera. The main issues that have been considered in order to relate the image and the scan coordinate systems are the image scale, skew and rotation using a simple 3-point affine transform.

The camera mounting plate allows the camera to be positioned at a known offset to the laser and then be raised and lowered with the CMM gantry to make the best use of the available image area. The webcam used has a flexible 'ball and socket' mount on a spring-clip base which allows for easy positioning of the camera, however it makes precise alignment of the camera axes very difficult to achieve by eye. The digital camera requires a screw fixing via the tripod mount on the base of the camera to the mounting plate.

#### **4.3.2.1 Camera Perspective Issues**

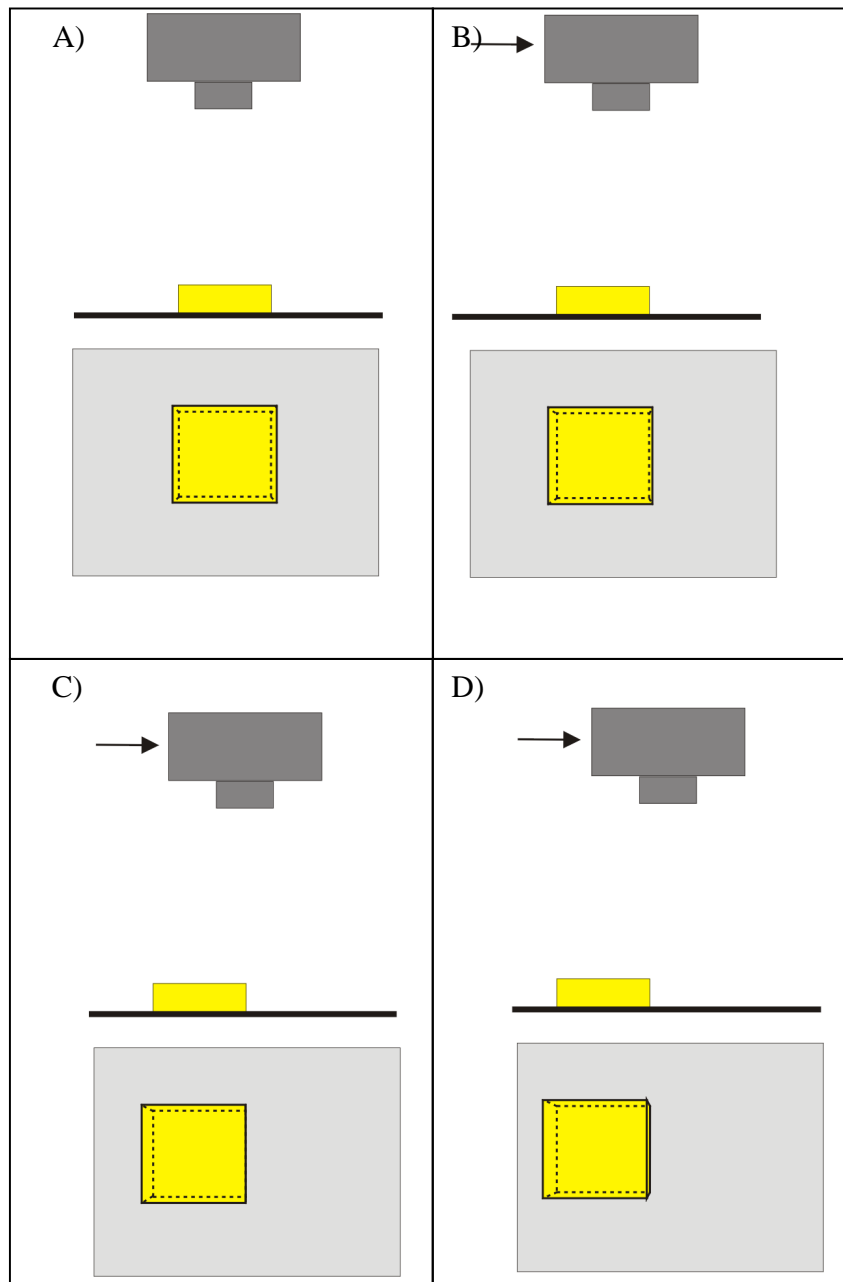
In a 2D camera image, by definition it is not possible to gauge the height of any edge. The human vision system imposes an impression of depth on real-world scenes that is 'learned' from visual cues such as perspective and context scale. The camera system alone does not have such intrinsic knowledge. In a 'synthetic' image all the edges of a shape appear at the same distance and only our experiences of the real-world and the skill of the artist give 'depth' to the picture. (Of course, the human vision system can also be fooled by optical illusions).

Within a camera image representing a real world scene surfaces that are closer to the lens appear larger than those in the distance. Within the context of this work this presents a problem, because the upper surfaces of object will appear larger to the camera than the calibration scale (recorded at the height of the scan bed) would recognise it to be at the base height.

Given that the object height is limited by the range of the laser (in this case 2cm) this error should be negligible for these experiments and, since the original intention was to capture regions of the object around the edges rather than precise edge locations, this was not determined to be a major issue. However, for object / laser combinations of greater height / range this issue may present problems

Camera perspective becomes more of an issue for the method involving the selection of orientation by nearest vector, especially in areas where there are corners. Selecting from a scan region of incorrect orientation because the edge position is misaligned by a couple of pixels is a real possibility.

There is also a related problem that edges further from the centre of the image will appear to be slightly offset from their 'correct' position towards the edges of the image. If the camera is placed directly over the object then this distortion will be distributed equally over the image, however if the object is offset in the camera's field of view then the effect varies relative to the position of the object. This effect means that it is possible in some situations to see vertical faces of the object as shown in Figure 4-17 even when the camera is oriented parallel to the scan bed.



**Figure 4-17:**

**A) The object is centred under the camera. Perspective distortions are distributed evenly.**

**B) As the camera is moved to the right the object appears displaced to the left. The left edge appears further from the true edge position and the right edge appears to shift closer to the true edge position.**

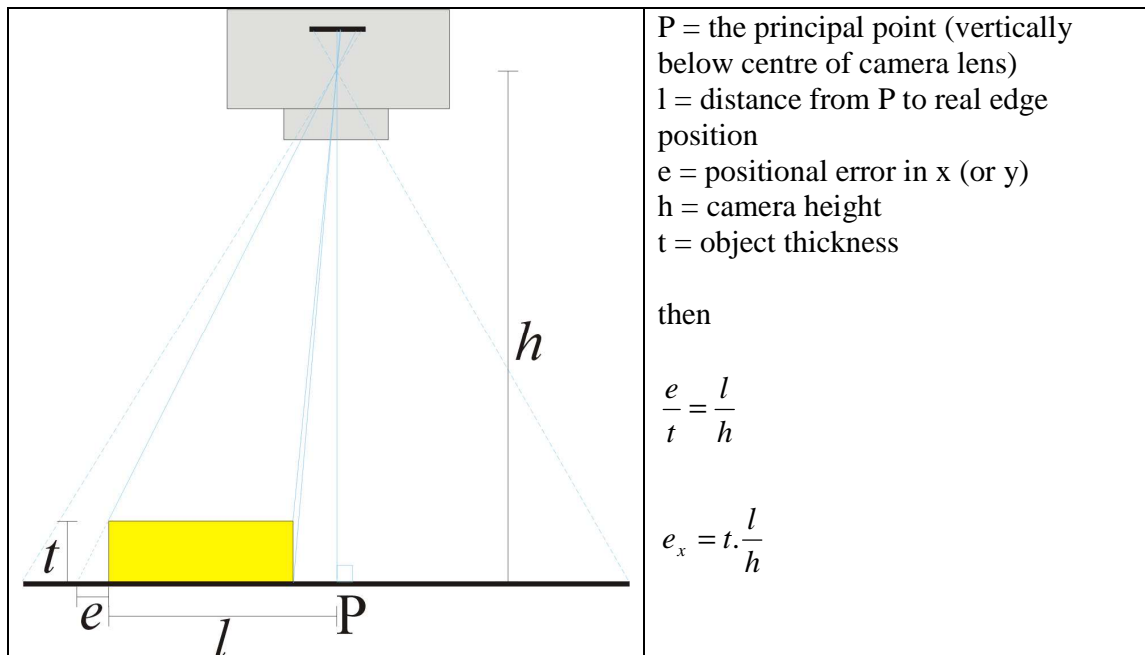
**C) The camera is aligned directly over the right edge of the object – this apparent edge position is aligned directly above the true position.**

**D) As the camera is moved further to the right the apparent position is shifted further left. It is now possible to see the right side vertical face of the object from the camera position.**

This combination of parallax (viewpoint) and perspective issues further complicates the task of relating the camera image to the scan coordinate system.

Positional errors due to the object height must be taken into consideration for a commercial system and if an approximate maximum height can be provided it may help with determining the object dimensions more accurately. In a first case, it should be possible to use calibration objects of known dimensions (including height) to provide a reference scale for the camera. (This would be particularly useful in conjunction with the idea for laser pre-scanning suggested in Chapter 8).

If the distance of the camera and the object height are known then it is possible to calculate the positional offset due to this effect by similar triangles.

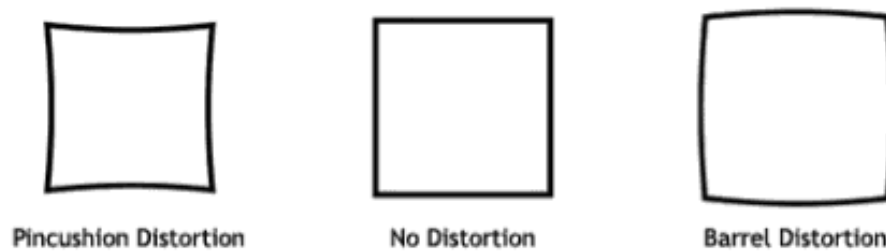


**Figure 4-18 illustrates parallax error in the camera. It is possible for the camera to ‘see’ vertical edges of the object as the distance from the principal point increases.**

Thus, for a camera height of 20cm, with an object thickness of 1cm and distance  $l$  of 5cm the error offset would be 2.5mm. Even at a fairly coarse resolution of 0.5mm per scan point this equates to an error of 5 scan lines which is likely to cause a positional error when scan regions are calculated for the real object based on the camera image (i.e. the scan region will be displaced from the intended location).

#### 4.3.2.2 Camera Calibration for Lens Distortion

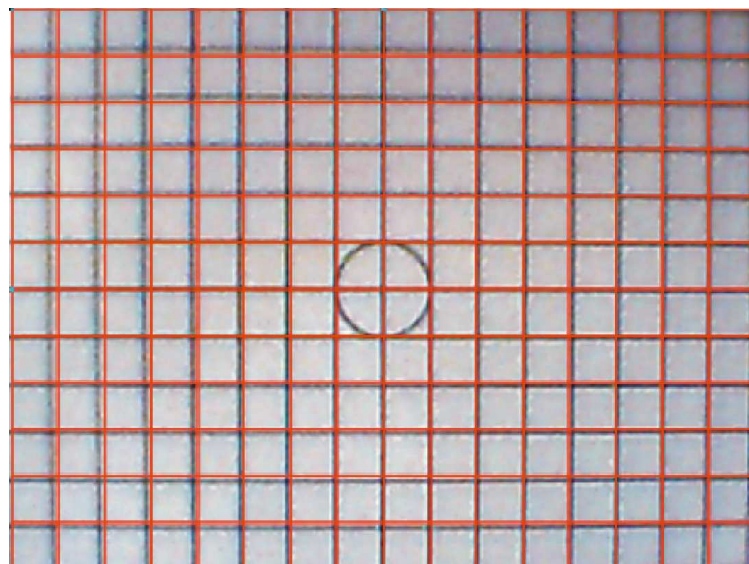
It is well-known that wide-angle lenses often suffer from barrel distortion, causing edges that are straight in the real-world to appear slightly convex in the camera image. Conversely, telephoto lenses suffer from the opposite effect, known as pincushion distortion, where straight edges appear concave. Both effects are especially apparent where edges are close to the edge of the image frame.



**Figure 4-19: diagrams showing pincushion and barrel distortion of edges.**

**These effects are associated with the lens characteristics of the camera.**

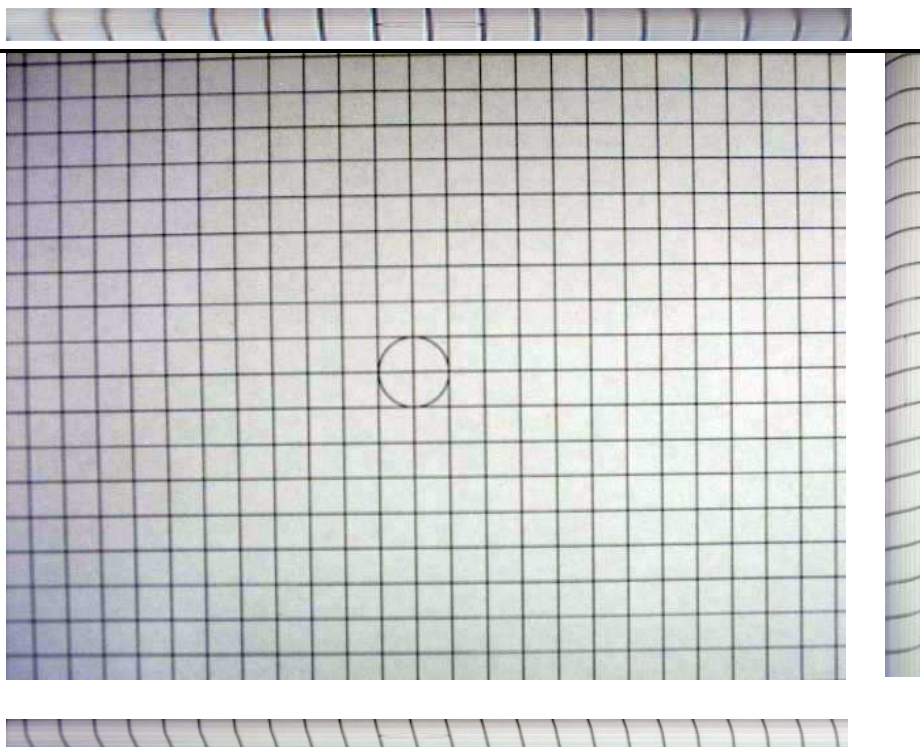
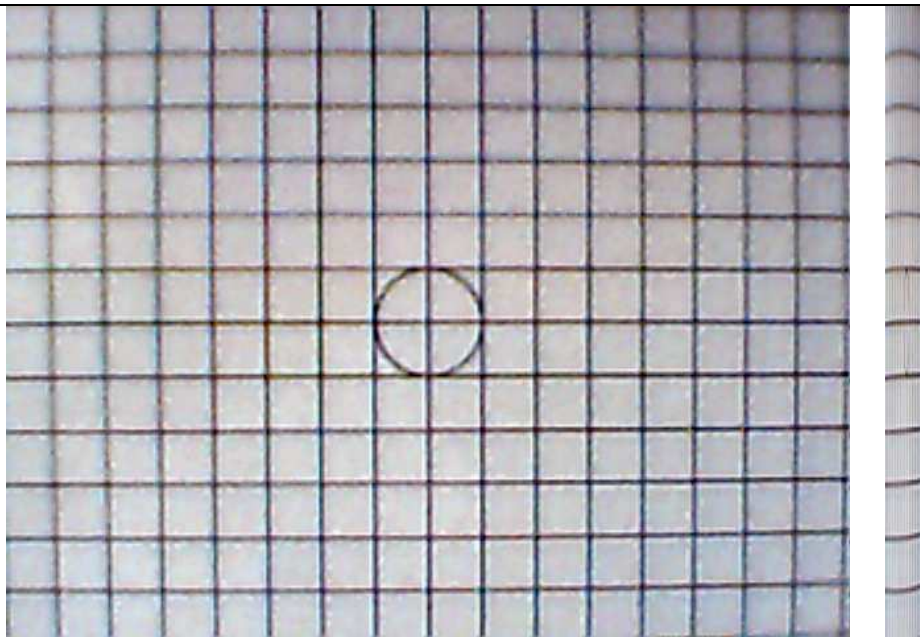
Such distortions are usually measured as the amount a reference line is bent as a percentage of picture height. For most consumer digital cameras pincushion distortion is lower than barrel distortion with 0.6% and 1% being typical values respectively. This is shown in Figure 4-20 below



**Figure 4-20 Webcam image of 5mm square grid compared with original grid pattern (overlaid in red) demonstrates lens barrel distortion.**

The webcam used in this experiment showed significant barrel distortion over the image area. The digital camera also displayed some barrel distortion although comparatively less than the webcam for the same height. This is shown in Figure 4-21 and 4-22. In both images the camera and image centre are aligned on the centre of the circle. The webcam image area is approximately 8cm x 6cm at a height of 16cm above the scan bed and displays noticeable radial distortion across much of the image. This distortion is more evident when the aspect ratio of the image is compressed by 95% in the x or y dimension. The lines toward the periphery of the image display considerable curvature compared to the lines in the centre of the image. (The tilt of these lines indicates the camera is also slightly rotated with respect to the grid and the lines are not parallel because the camera's image plane is not exactly parallel to the plane of the scan bed).

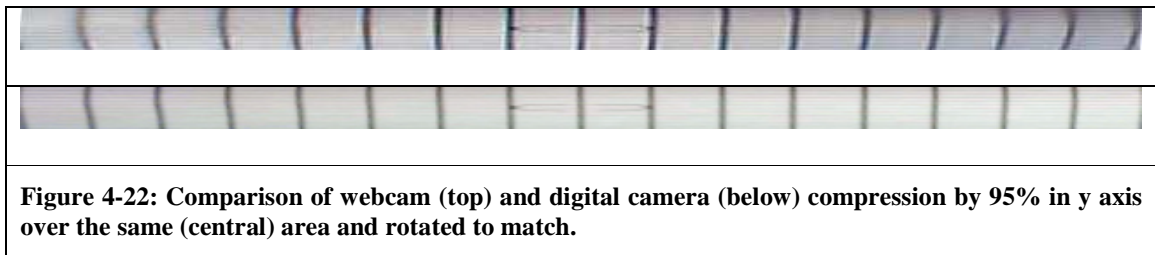
It was hoped that radial distortions would not cause a serious problem as the idea is not to find the precise edge location but to build scan regions around the edges. This proved to be a more complicated issue than originally presumed and any commercial implementation of this work would certainly require some means of correcting such distortions, however due to time constraints these methods have not been implemented here. These distortions are well-known and numerous correction methods have been presented [114].



**Figure 4-21 Webcam (top) and digital camera (bottom) images of 5mm square grid at camera height 16cm. Curving of the square grid is evident in the image. This is more obvious when the image is compressed in the x or y axis (displayed below and to the right of the main images). The digital camera displays less distortion at the same camera height, despite covering a larger area.**



The digital camera also displays some barrel distortion but it should be noted that at the same image height the image area is approximately 50% larger than the webcam. When the digital camera image is cropped, scaled and rotated to match the webcam then it can be seen that the digital camera is less distorted by radial effects over the central area of the image, as shown in Figure 4-22 below.

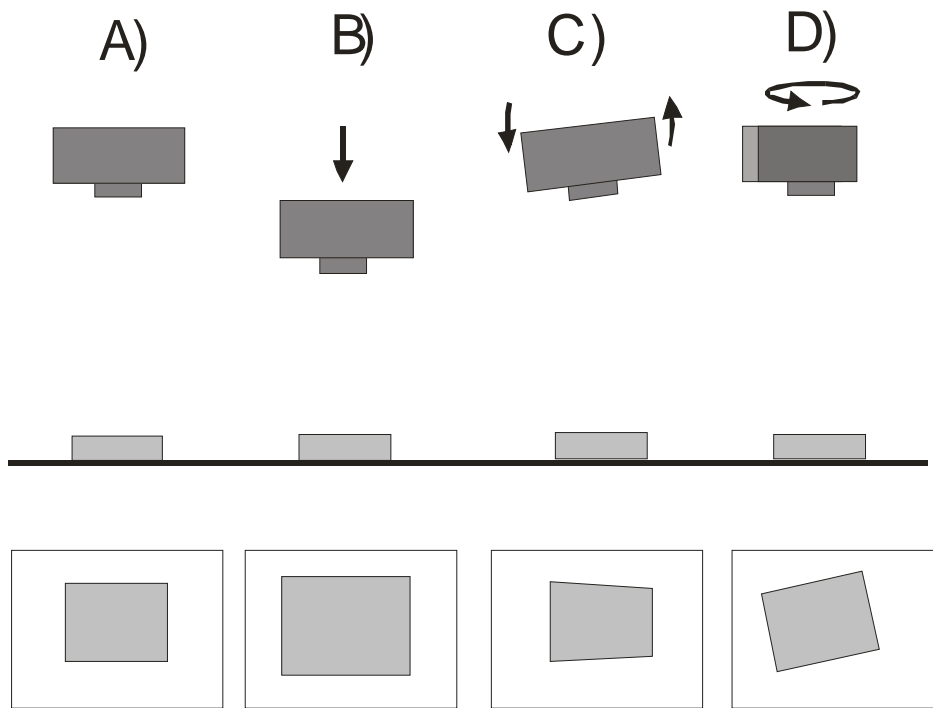


Furthermore, it can be seen that the webcam also displays chromatic aberration and poor focusing across the image plane. Aberrations in the camera lens cause non-linear distortions that are not simple to correct.

### 4.3.3 Simple Corrections for Camera Scale, Skew and Rotation

Initially a simple 3 point calibration system was used, which is sufficient to provide a basic transformation between the reference and target images. An affine transformation model has been applied, incorporating image scaling, skew and rotation components in order to calibrate the scan space and the image space relative to one another.

Image scale refers to the change in the apparent size of an object when viewed from different heights at the same image resolution as indicated in Figure 4-23 by the change from A) to B). The higher the camera, the smaller the image area the object occupies. Image skew is the orientation of the image plane to the scan bed. If the camera image plane is not parallel to the scan bed then any object placed under it will be distorted, with the edges closer to the camera appearing to be larger (as shown in Figure 4-23c). The distortion shown is only in one axis – in practice there may be a combination of tilt in both the x and y axes. Image rotation is the amount the axes of the image space are rotated from those of the scan space as shown in Figure 4-23d).



**Figure 4-23 shows how changes in camera position affect the image of the object.**

**A) Provides a reference where the bed is square to the image plane.**

**B) The camera is moved closer and the object appears larger in the image.**

**C) Demonstrates perspective skew due to the camera tilt in one axis.**

**D) Shows the effect of rotating the camera.**

#### 4.3.3.1 Implementation of Simple Affine Transform

An affine transformation takes any coordinate system in a plane into another coordinate system that can be found from such a projection. Under affine transformation, parallel lines remain parallel and straight lines remain straight. The mathematical principles of coordinate system transformations are well-known and the reader is referred to Foley *et al.* for a more detailed explanation of the theory [115].

A 2D coordinate system is defined by an origin, two (non-parallel) axes and scale factors for each axis. This can be described by 3 points, one for the origin and one for the unit distance along each of the two axes. It takes six numbers to specify three points.

For three points:  $(x_1, y_1)$ ,  $(x_2, y_2)$  and  $(x_3, y_3)$ , then given an affine transformation as above, the corresponding three transformed points can be found from:

$$\begin{array}{lll} x_1' = a*x_1 + b*y_1 + c & x_2' = a*x_2 + b*y_2 + c & x_3' = a*x_3 + b*y_3 + c \\ y_1' = d*x_1 + e*y_1 + f & y_2' = d*x_2 + e*y_2 + f & y_3' = d*x_3 + e*y_3 + f \end{array}$$

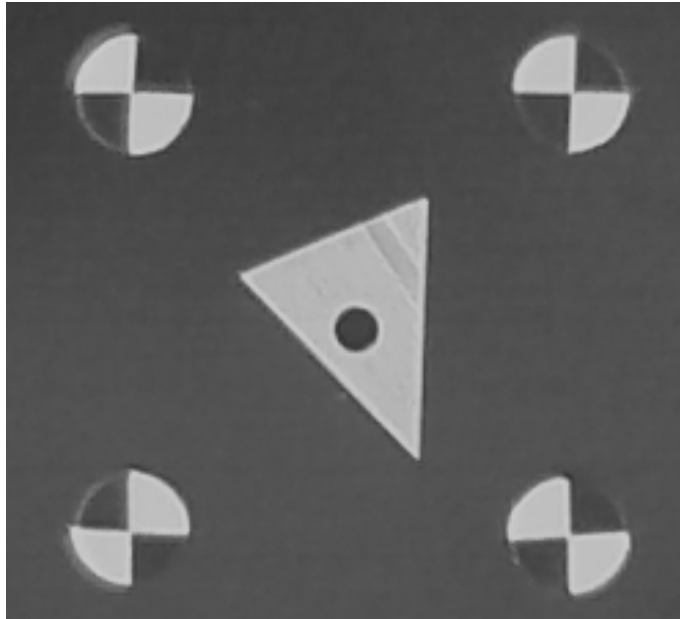
Conversely, if the three points in the transformed (primed) coordinate space are given (corresponding to the three unprimed points), the above set of equations can be solved for the six coefficients. These coefficients can then be used in to transform any point in the original coordinate space to its location in the primed coordinate space.

Therefore, for this project, starting with 3 points specifying the known points in the reference coordinate space and the 3 points that specify the corresponding positions in the target coordinate space, the values of a-f can be calculated. Then the entire image can be transformed point-wise as follows:

$$\mathbf{x}' = \mathbf{a}*\mathbf{x} + \mathbf{b}*y + c$$

$$y' = d*\mathbf{x} + e*y + f$$

Due to the global nature of the affine transform it cannot be used to model local deformations, however it was considered that this would be sufficient for the requirements of this system.



**Figure 4-24: Calibration markers placed around scan object. The markers are placed at known scan coordinates and the corresponding image pixel coordinates at the centre of each marker are recorded. Note the origin marker (bottom left) is rotated 90 degrees to the other markers for easy identification.**

In order to find the known points in the image space calibration markers (small circular disks painted with a black and white quarter pattern) were placed at known coordinates on the laser scan bed around the object to be scanned, using the laser itself as a guide to positioning the markers. These coordinates marked the (zero x, zero y) position, the (max x, zero y) and (zero x, max y) positions of the scan.

Once the markers were placed around the object the camera was positioned over the object and the height adjusted until all the calibration markers were visible in the camera's field of view. The camera (target) image was then captured as a 640x480 pixel digital image using the basic software provided with the webcam.

Initially an attempt was made to recognise, locate and register these calibration marks in the target image automatically, however this proved only partially successful (in that it was not precise enough to determine the intersection of the centre of the calibration markers) so a simple manual calibration system was developed that simply required the image pixel (x, y) position of the centre of each marker to be recorded and entered into a dialog box along with the corresponding position in the reference (scan) image by the operator. The x,y

position of any image pixel can be read in the status bar of the test program simply by hovering the mouse over the requested position.

The calibration dialog window is titled "Select Number of Calibration Points" with a dropdown menu set to "3". It contains three main sections for calibration points (CP 1, CP 2, CP 3), each with a "CP #" label and a 2x2 grid of input fields for image and scan coordinates. Below these is a section for "Calibration Point #" with labels for "Image Coordinates: X,Y (Pixels)" and "Scan Coordinates: X,Y (mm)". At the bottom, there are input fields for "Laser Scan Height" (50 mm), "Laser Emitter Detector Separation Distance" (40 mm), and "Max Edge Height" (20 mm). A "Laser Zero Orientation" section has two radio buttons: "Parallel to X Axis" and "Parallel to Y Axis". At the very bottom are three buttons: "Cancel", "Clear Points", and "OK".

CP #	Image X (Pixels)	Image Y (Pixels)	Scan X (mm)	Scan Y (mm)
CP 1	145	407	100	100
CP 2	521	404	160	100
CP 3	138	92	100	150

Calibration Point #

Image Coordinates: X,Y (Pixels)

Scan Coordinates: X,Y (mm)

Laser Scan Height: 50 mm

Laser Emitter Detector Separation Distance: 40 mm

Max Edge Height: 20 mm

Laser Zero Orientation:

☐ Parallel to X Axis

☒ Parallel to Y Axis

Buttons: Cancel, Clear Points, OK

**Fig 4-25 Calibration dialog window showing corresponding image and scan coordinate pairs**

Once the registration data has been entered the affine transformation coefficients are determined and stored with the image to convert any point any point in the reference space to the corresponding position in the target space (and vice versa, however only the transformation from scan space to image space is currently used).

In practice however, the 3-point affine transform is limited in its ability to correctly calibrate the camera image as there is an inherent assumption that any distortion, scaling or perspective correction is linear over the extent of the image. In an ideal situation this would be the case, however in practice the radial distortion and the relatively poor quality of the camera optics render this assumption invalid.

## **4.4 Other Systemic Distortions**

The hardware and scanning software used in testing the hypothesis proposed in this thesis has displayed a number of technical difficulties in producing reliable (and reproducible) scans during testing. These problems fall into 2 main categories:

**Systemic Noise** consisting of unavoidable noise, such as vibrations caused by movement in the system, ‘whiplash’ in the movement of the CNC axes and (potentially) avoidable noise, caused by poor quality signals on various data lines.

**Timing Issues** caused by problems with task scheduling, which proved to be a serious concern in producing useable scan data.

### **4.4.1 Synopsis of Hardware Operation**

Scan3D and the CNC software (AxController task) run as two different tasks under Windows. The laser runs independently of these tasks via an analogue card. The data is read and transferred from the laser analogue card to Scan3D one line at a time. The laser takes samples continually but these data points are only stored from the point at which a digital (timing) signal goes high.

Scan3D initiates the scan process by instructing the AxController process to send a move signal to the controller hardware, prior to instructing the laser card to begin storing data samples on a timing signal. With this model of laser, data is only recorded unidirectionally (i.e. the laser records data whilst moving from left to right, then returns to the initial position without recording data on the right-to-left move).

The sample-rate of the Matsushita laser used is 1 point per millisecond. A number of sample points are averaged to provide each data point in Scan3D (typically 4 or 5 samples per scan point depending on the speed of the movement of the CNC machine and the scan resolution). Each data point is read by the laser and stored on the analogue laser card until Scan3D sends a message to tell it to stop at the completion of the scan line. The whole line of data is then transferred to the PC and stored in a data array in Scan3D. The sample rate of the laser and the velocity of the scan head are known, so for the pre-defined scan area the number of samples required can be calculated. Any excess samples recorded during the deceleration period of the laser are discarded.

The software collects data from the laser continually, based on an initial timing signal from the laser. The laser must be accelerated to a constant velocity by the CNC machine before samples are recorded as only the speed of the scan head and the sample-rate are taken into consideration in determining the position of each scan point relative to the scan start point for each line. If samples were recorded while the laser was accelerating the samples would be unevenly spaced. Once the laser is up to speed the CNC sends a timing signal to Scan3D via the controller card which initiates the data collection. Data is sent continually, even after traversal of the current scan line has been completed, until a 'stop sample collection' signal is received from Scan3D.

Because the area of the scan is pre-defined and as the speed of movement and acceleration rate of the CNC machine are known, the time delay between the initial 'move' command and the timing signal can be calculated. The timing of this signal is critical to the accuracy of the scan and has proven to be problematic due to two identified problems described below:

#### **4.4.2 Timing Signal Noise**

A significant level of noise was present on the line from the CNC machine to the controller card. This noise level occasionally reached the trigger level that would initiate the laser to start recording data samples earlier than expected. Because the only the initial x,y coordinate for the scan line is 'known' (and all subsequent points are calculated relative to this point) Scan3D assumes that the first data recorded was at this point. However, as the laser sampling was triggered early, the data line is shifted to the right as the first point recorded is assumed to be the correct position. At the other end of the scan line, 'good' data is discarded because only the required number of data samples are converted in Scan3D to fill the scan line array.

This issue has been compensated for by placing a suppression capacitor on the signal line to the controller (i.e. across the digital input to ground). This has the effect of increasing the signal to noise ratio of the line, which has effectively removed the problem since the issue was identified. The capacitor does incur a small delay in the edge response of the timing signal, but this is quantifiable and can therefore be compensated for if necessary.

### 4.4.3 Operating System Latency

AxController and Scan3D run as separate processes and both are under the control of the Windows task manager, along with any other processes running on the PC. This can cause issues when Windows switches tasks.

Scan3D instructs AxController to begin movement and send the signal to the laser analogue card prior to the instruction to the laser card to begin storing samples. If Windows does not return control to the Scan3D process after the AxController process has sent the 'Move' signal to the CNC hardware then the timing signal can be received too late. (i.e. by the time Scan3D receives its next timeslice from the operating system and signals the laser to begin recording data the CNC machine has already started to move the laser).

The degree of latency is variable (anything from 1ms up to 1s, depending on current activity on the PC). Because of the late start, when the scan line data is stored in Scan3D it is effectively shifted to the left because it is missing the correct sample points for the start of the array.

This second issue is more problematic as it is an inherent problem with the software design and its interaction with the operating system. An attempt to compensate for the problem by placing a 'Wait' timer in the AxController software has proved to be mostly successful at the expense of a delay in the start of each scan line. Adding a 0.5 second delay to each scan line does not at first sound like much, but when each scan comprises of hundreds or thousands of scan lines it adds a significant amount of time to the overall scan; however as it is currently not possible to adjust the individual scan lines so the only 'fix' when this problem occurs is to repeat the entire scan, the addition of the 'wait' timer is the lesser of two evils.

Unfortunately, despite the addition of the wait timer, this problem still happens occasionally and unpredictably. It seems the best policy is to close down any unnecessary tasks running on the test PC to reduce the load and minimise any activity on the PC whilst the scan is active. N.B. this problem only manifested itself when Scan3D is used with the 'new' Axiomatic controller which uses USB rather than a serial bus. Because USB is so much faster than the serial bus, in effect the old controller had an inherent 'wait' timer built-in to the delay in transferring the data via the serial bus. Ironically, it is the improved speed of the controller hardware that has caused this timing issue to manifest itself as a problem.



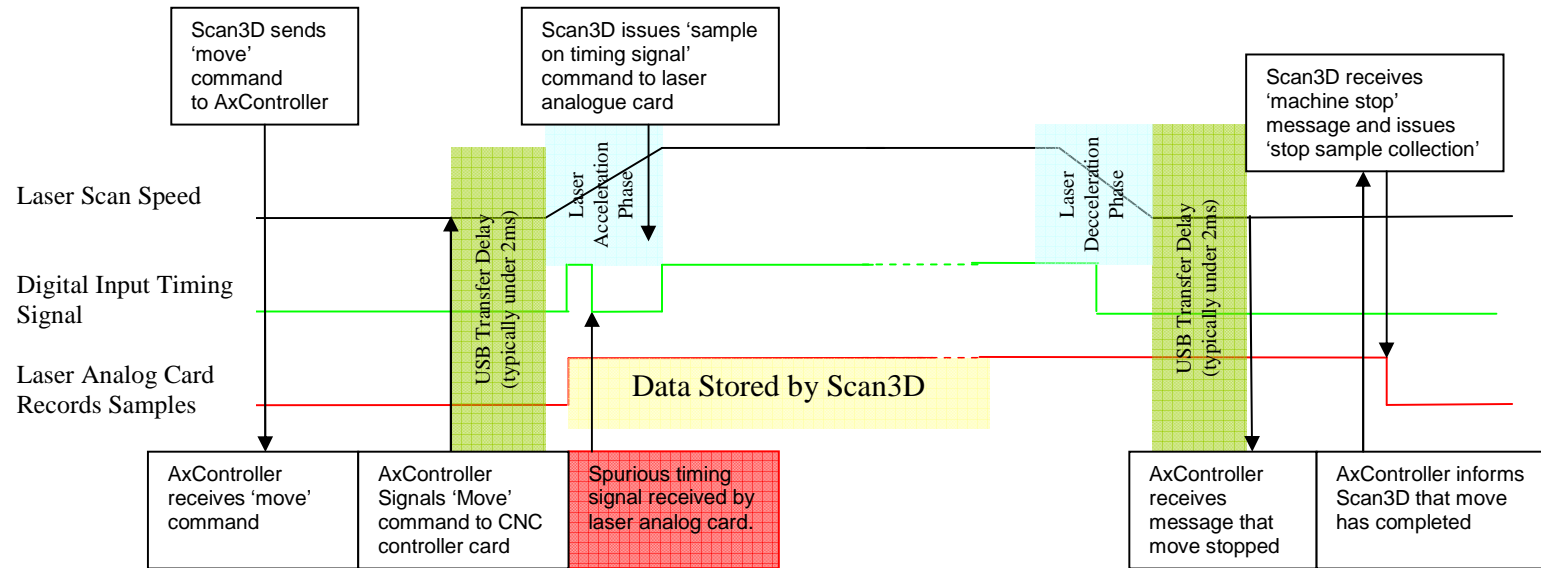
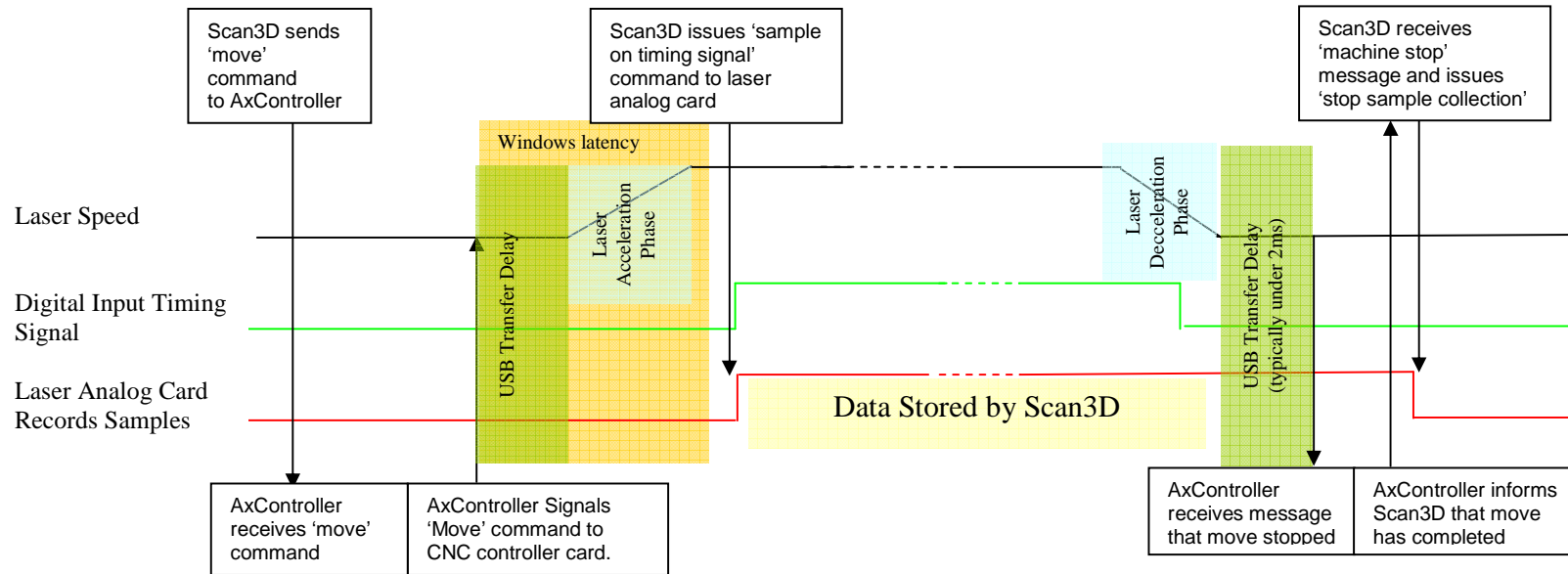
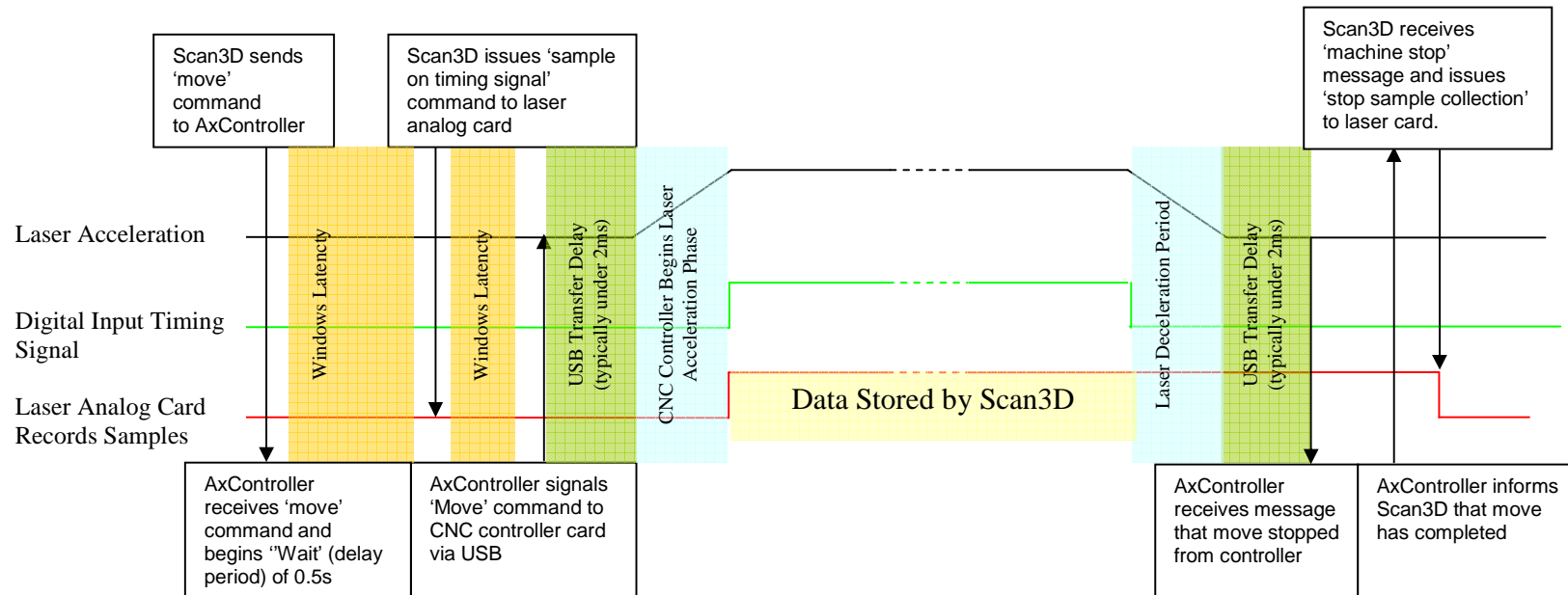


Figure 4-26a) Noise spike on the digital input timing signal line causes the early trigger of data collection by the analogue card



**Figure 4-26b) Latency in task switching causes a delay in issue of digital input timing signal resulting in late start of data collection**



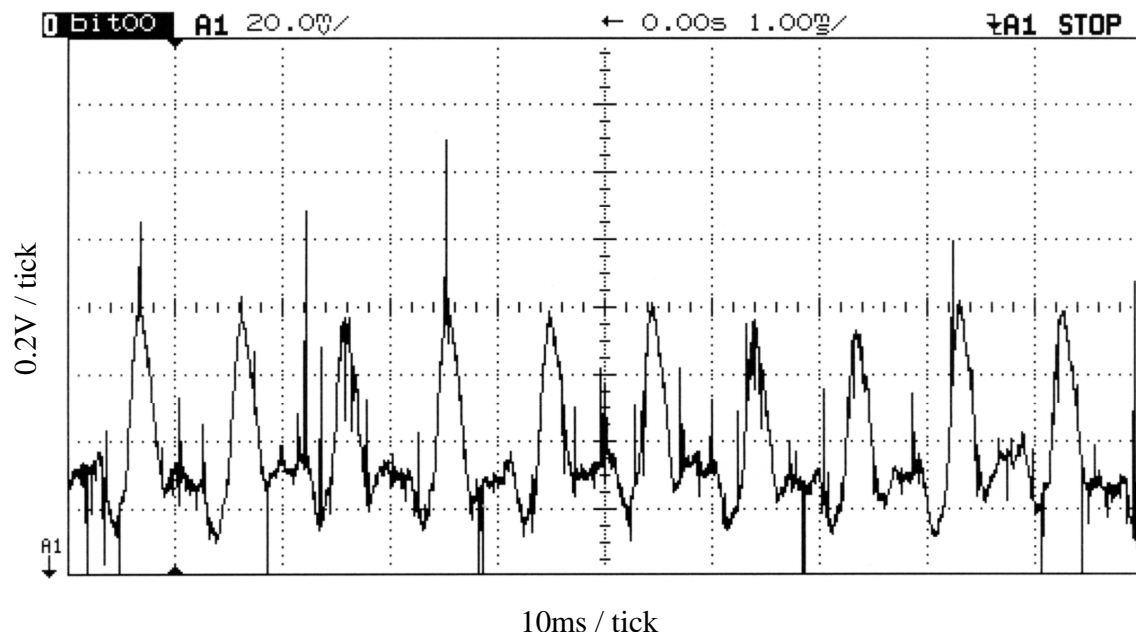
**Figure 4-26c) Corrected timing diagram for Scan3D and AxController tasks with relation to the movement of the laser gantry of the CNC machine: addition of the wait timer in the AxController process ensures that Scan3D will have sufficient time to send the timing signal to the laser analogue card**

#### **4.4.4 Noise on the Laser Analogue Data Line**

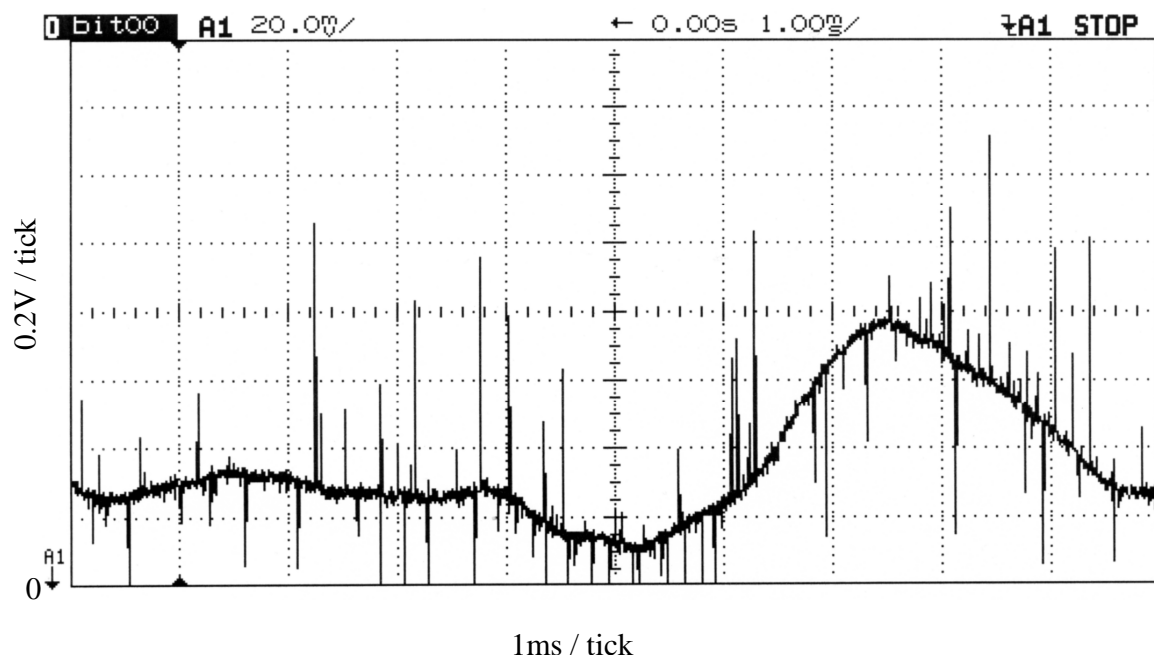
As well as the noise on the timing line it was discovered that there is also background noise on the laser analogue data line. This noise is a  $\pm 0.3V$  amplitude wave of approximately 90Hz cycle. This is complicated by additional random spike noise of up to about  $\pm 0.7V$ .

The full range of the laser is  $\pm 5V$ , covering 20mm, so a 1V 'spike' in either direction would be as much as  $\pm 2mm$  deviation from the 'true' value at that point. As Scan3D averages a number of samples from the laser per value of output, in practice the overall effect is somewhat less than the 'worst case', but the output value at the same x,y position can often vary by more than  $\pm 0.5mm$ , which is unacceptable when trying to measure to an accuracy of 0.5mm, which is a fairly coarse scale of accuracy for many real-world applications.

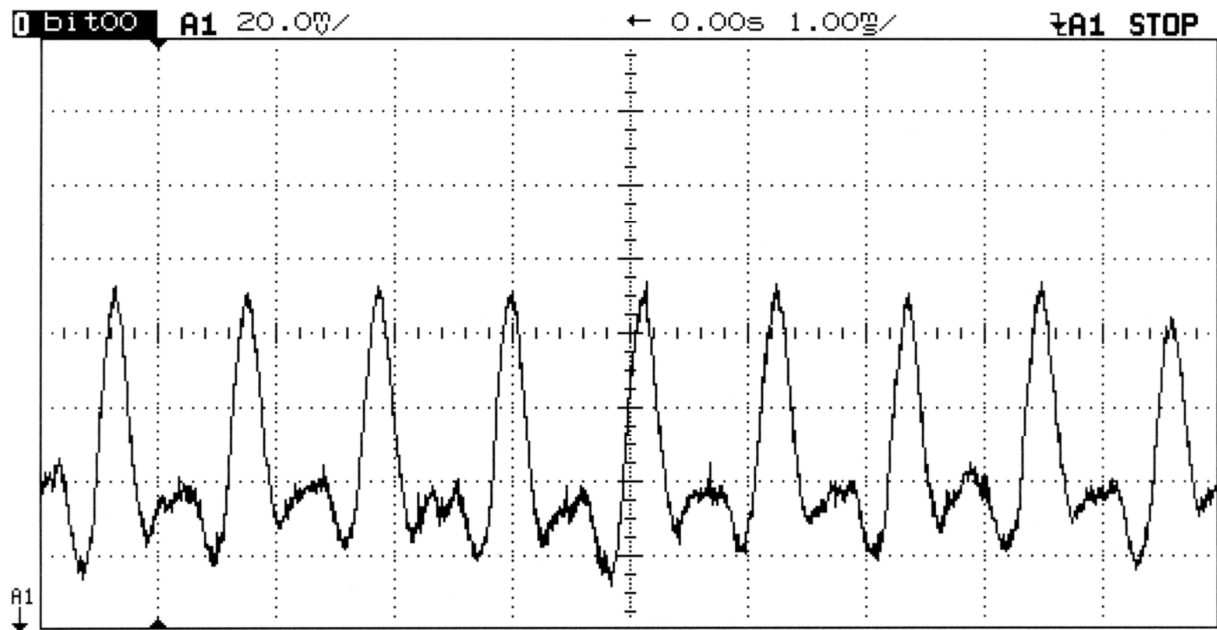
This noise is especially evident when trying to combine files from a number of orientations, as even the base-level height readings over the scan bed are different from file to file. Even two scans of the same object at the same orientation display a significant level of hysteresis, because even though they start and end at the same point it is almost certain that the two scans will be 'out of sync' with respect to the wave noise.



**Figure 4-27a): Noise on laser data signal.** This trace shows a wave form complicated with noise spikes (see below). The variation in the signal level is over 0.5V and the spikes further increase this effect.

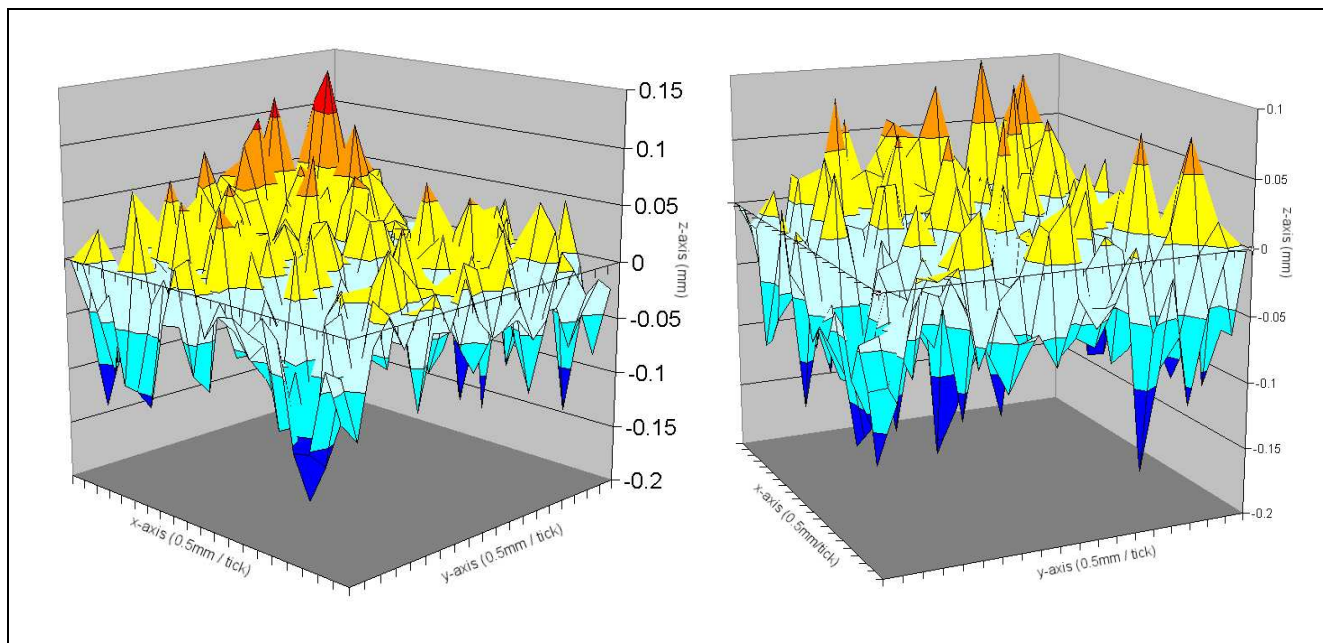


**Figure 4-27b) Enlargement of single noise wave cycle showing spikes of variable magnitude and unpredictable frequency on the laser data line.**

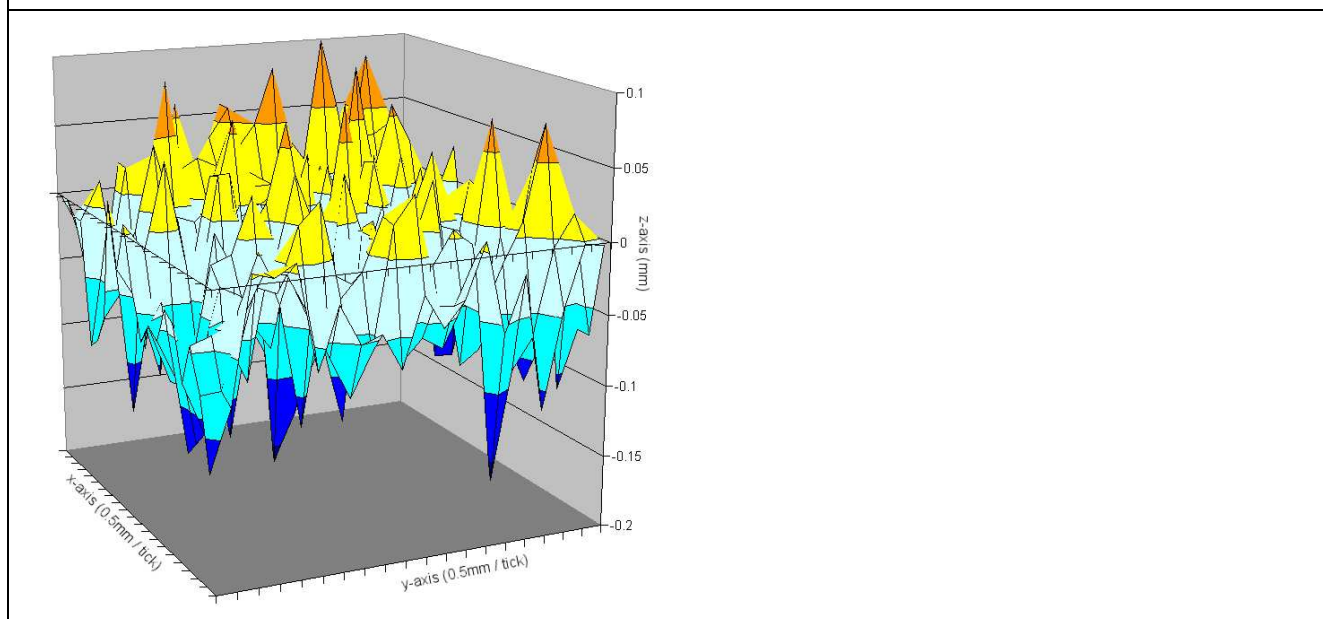


**Figure 4-27c) Addition of a capacitor across the laser data signal to ground has removed the spike noise, however the wave noise remains in the system.**

Removal of the spike error was achieved by placing a suppression capacitor on the data line – this effectively removed the worst of the erroneous values, however the waveform error still remains in the system, causing variations in the height reading of the laser of approximately  $\pm 0.3\text{mm}$  (i.e. a range of over  $0.5\text{mm}$ ). Unfortunately a capacitor large enough to suppress the ‘ripple’ wave will also suppress detail in the scan resolution. However the width of each ripple cycle encompasses 9 data sample points (assuming 1 sample per millisecond) which will be averaged to a single data output value in Scan3D. This averaging effectively hides the sine wave of the ripple, however the ripple wave does add a significant level of scatter to the data, which will be visible as a base level of hysteresis in the scan data as shown overleaf in Figure 4-28.



**Figure 4-28: Typical variation in values caused by signal noise on the sensor data for two different scans of the same area of flat surface of scan bed with the same laser orientation. The difference between the scans is shown in the graph below, showing that some variation in values for the same point in different scans remains in the system.**



## 4.5 Summary

This chapter outlines the hypothesis for the work in this thesis: that by locating the edges of the object from a camera image it is then possible to relate their position to the laser scanner, and orient the sensor relative to the orientation of the edges according to the chosen scan method.

Three methods are suggested for development. In Method 1 the dominant scan direction is chosen for a single scan. This method provides an ‘optimal’ *single* scan but will not remove all errors if there are any edges at other orientations. Method 2 suggests the development of partial scans based on the discovered edges. Two potential approaches are described: firstly using ‘scan orientation regions’ grown around the discovered vectors and secondly by determining the orientation of only the nearest vector to each point in the image. Method 3 would require only a single scan, with the laser sensor continually rotated to the optimal angle for the current scan position, however this method is not currently practical due to limitations of the available hardware.

A number of hardware and calibration issues have been identified that may cause problems with the effective operation of the above methods. These are related to the alignment of both the laser and the camera mounting and distortions related to the intrinsic parameters of the camera lens. The affine perspective transform is not sufficient as camera distortion is non-linear leading to incorrect calibration and difficulty in accurately matching points between the co-ordinate systems.

Incorrect alignment of the laser will cause the measured points to deviate from the CCM coordinates. Tilting the laser may also cause some scaling error due to the slightly increased distance from the laser to the measured point. This problem becomes particularly apparent when regions from multiple scans at different laser rotation positions are combined.

Whilst the best solution is to avoid the problem in the first place by correct alignment of the laser, it is likely that some residual error will be present in the scan positions, as it is extremely hard to verify the exact position of the laser spot on the base due to the apparent spread of the spot. Without a precise means of measurement it is impossible to verify the position to much less than 0.5mm by eye alone.



Compensation for the alignment error mentioned above is difficult as the degree of error is variable according to the height of the laser above the scan point (and the true height of any point measured above the scan base). Compensation for the laser tilt error is also possible (within the limits of the operator's ability to re-centre the laser spot on the same zero-position).

In both cases, the calculations to correct the error are dependent on the ability to measure the error to a degree of accuracy at least equal to that used in the scanning process itself. Failure to properly calibrate the laser scanner will result in misalignment of the scan regions when multiple scans are merged together, resulting in discontinuities in the scan data where these regions meet. Even a slight error of 0.5mm in the alignment of the scan regions can appear as a noticeable discontinuity at the scales which the objects are often being measured – especially where the data describes a geometric edge.

The accuracy to which the laser can be aligned is also limited as, (currently at least), the only means of alignment is by eye. Where the laser hits the scan bed the spot of light spreads out, making it very difficult to align the laser to sub-millimetre accuracy (with respect to any 'useful' resolution in scanning). A 2D photocell in the scan-bed with a resolution similar to that of the laser could be used for calibration, however this option is not available in the current test rig. Timing and systemic noise issues related to the operation of the hardware and the operating system have also been identified and have been addressed as far as is practical.

It is realised that this technique will not find a solution to all the possible occlusion errors for all objects. Specifically, there will be problems with certain corner features that will cause secondary reflection issues no matter which direction they are scanned, however this is a drawback of the triangulation laser not of the technique, which should still 'flag' the position in the image as an area of concern where the data may not be as accurate as might be hoped.

Methods will be required to identify regions in the optical image that correspond to positions where laser triangulation distortions may occur. Then the laser scanning process can be improved and errors minimised by re-orienting the scanner so that it approaches such areas at an optimal orientation. Conversely, the identification of regions of low complexity will allow them to be scanned more quickly.

- 
- 106 Maekynen A. J., Rahkonen T. E. and Kostamovaara J. T., '*CMOS Binary Position-Sensitive Photodetector (PSD) Array*', Proceedings of SPIE, Sensors, Sensor Systems and Sensor Data Processing, vol. 3100, pp. 101-109, September 1997.
- 107 Xu Y., Tang J. and Zhong W., '*New Method of Processing the Signals of a Position Sensitive Detector*', Proceedings of SPIE, Advanced Photonic Sensors: Technology and Applications, vol. 4220, pp. 260-263, October 2000.
- 108 From Matsushita Technical Document available at [www.matsushita.es/download/sensor/LM10\\_cat\\_en.pdf](http://www.matsushita.es/download/sensor/LM10_cat_en.pdf)
- 109 Amann M. C., Bosch T. M., et al., '*Laser Ranging: A Critical Review of Unusual Techniques for Distance Measurement*', Optical Engineering, vol. 40, no. 1, January 2001.
- 110 Poliakoff, J.F., private communication to Author. 2006 (See Appendix A)
- 111 Tsai, R., '*A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf TV cameras and lenses*'. IEEE Journal of Robotics and Automation, 3(4):323–344, August 1987.
- 112 Faugeras O.D., Luong Q.-T., and Maybank S.J. '*Camera self-calibration: theory and experiments*'. In European Conference on Computer Vision, pp 321–34., 1992.
- 113 Faugeras, Olivier. '*Three-Dimensional Computer Vision*', MIT Press, 1993.
- 114 Tordoff, B.J. & Murray, D.W., '*The impact of radial distortion on the self calibration of rotating cameras*', Computer Vision and Image Understanding 96 pp.17-34 (2004)
- 115 Foley, J.D. *et al.* 'Introduction to Computer Graphics' Ch. 5, pp.168-177, Addison Wesley, MA, 1994

## **Chapter 5 – Edge Detection and Vectorisation Methods for Improved Scanning**

As discussed in Chapter 3, many methods exist that can be used to detect features in digitised images. These methods have relevance for this project because there is a requirement to detect the positions of geometric edges (and changes in reflectivity) that are often responsible for the occurrence of errors in the scanning process when a single-perspective laser sensor is employed (as discussed in Chapter 2).

The intent in this project is to determine the position of edges within images captured using a digital camera and to develop ‘scan orientation regions’ around those edges that will reduce the number of locations at which the laser sensor is at an ‘incorrect’ orientation relative to the geometry of the object. The concept of the scan region development is covered in Chapter 4.

In this chapter the means by which the edge detection and vectorisation algorithms were developed is discussed and an evaluation of the results of applying these algorithms to images captured using the test cameras is presented.

## **5.1 Image Processing Algorithms**

A selection of established edge detection algorithms was implemented using a common code platform in order to compare them in a controlled manner and provide unbiased results. The criteria for success are as proposed by Canny for his detector (see Section 3.3.3), with the additional requirement that the speed with which the results are obtained should also be a consideration. Whilst it was not expected (given the processing speed and memory available on modern PCs) that the algorithms would take an excessive amount of time to execute for a 'standard' sized image of the whole object compared to the time taken to scan the object, determining the relative execution time for each algorithm is a useful measure in evaluating the resulting image. The execution time of the image processing stage would be more critical if the edge detection was being done on a 'just in time' basis (See Chapter 8.4.3) and it is therefore pertinent to compare this factor for future use.

### **5.1.1 Greyscale vs. Colour Images and Image Resolution**

Most of the work on edge detection has been performed on 8-bit greyscale images: i.e. 1 byte is used to represent the intensity at each pixel position, yielding 256 levels of grey from black (0) to white (255). Colour digital cameras are now common and relatively cheap products and colour images appear to provide a greater level of information within the picture. There are several digital colour models available, the most common types being the RGB and CMYK models. These use 3 or 4 bytes to represent the value of each pixel: in the case of the RGB model one byte is used to represent the intensity in each of the red, green and blue planes that are combined at each pixel to give an overall colour and intensity.

Many colour edge detection methods attempt to extend the standard 'greyscale' edge detectors into the colour model by determining the edge map for each colour plane and then providing some method by which the output from each layer is fused into a single edge map, often employing a logical 'AND' or 'OR' operator across the image layers at each pixel position to determine the location of edges. Relatively few colour detection methods apply methods that do not involve some variation on this principle. One notable exception is the Generalised Compass Operator [116] which appears to show promising results.

Since color images provide more information than grey value images, more detailed edge information might be expected from color edge detection, however Novak & Shafer [117] found that 90% of edges are approximately the same in grey value and in color images, although it is possible that the remaining 10% may make a significant difference to the overall edge continuity. Although colour image output was available from the digital cameras, it was decided initially to implement several common edge detection methods using greyscale images to test the hypotheses presented in this project due to the complexities of the colour edge detection methods.

For this project it was often necessary to spray-coat the objects being scanned and imaged in order to reduce the chance of specular reflections occurring in the scan data. The sprayed surface means that images are generally monochromatic so there is no advantage (at this stage) in using colour images.

All images were therefore converted to a greyscale representation prior to edge detection. If the greyscale edge detection does not produce images of sufficient quality to determine the edge location and orientations then testing colour image edge detection and segmentation methods remains as a direction for future development that may improve on the results of the edge detection methods as implemented here.

The resolution of modern commercial digital cameras is generally in the order of several thousand of pixels in each axis of the image, between 2 and 4 million pixel cameras are commonly available. Webcams tend to have a lower image resolution, as the images they produce are designed to be transmitted over the Internet in 'real time'. The maximum resolution of the webcam used in this project is 640 x 480 pixels.

Using a high resolution image would significantly increase the processing time required to determine the locations of the edge vectors in the image, so the lowest acceptable resolution for the given scan resolution is used whilst remaining sufficient to define the edges accurately and provide adequate mapping between the scan and image.

The CNC machine used in this project is capable of scans of 0.01mm accuracy and for scans of this resolution it would be necessary to use a higher resolution camera image in order to provide an image space of adequate detail, however experiments show that the 640x480 webcam image resolution provided an acceptable level of detail in the captured

image when used with a test scan resolution of 0.5mm. Although much higher resolutions are possible the digital camera image was also limited to 640x480 to provide an ‘even’ match.

### **5.1.2 Software Framework**

Software has been developed to enable the edge detection and vectorisation algorithms based on the ‘Paintlib’ open source image-handling library [118]. This includes ‘Piclook’ by Bernard Delmee, which is a basic application that allows viewing, file-handling and a number of basic image processing operations to be performed on several commonly used image formats, however no edge detection or vectorisation functions are provided. Microsoft Visual Studio .NET was used throughout the development of the software to produce the test executable program.

The software used in this was developed by the author in order to provide a fair test of the various edge detection algorithms on a common platform (i.e. to ensure that the underlying image structure and convolution algorithm is handled in the same way for all the detectors). Commercial packages such as Matlab could be used however these packages are expensive to license (any target software would require licensing) and the internal working of existing routines is hidden, so it is therefore unknown whether the algorithms provided by such a package are appropriately (or equally) well-optimised, thus compromising the timing comparisons.

The software developed for this project makes use of the paintlib library as a basis for image-handling. Paintlib is copyright 1996-2002 Ulrich von Zadow (and other contributors), however the library is freeware and all the code relating to the implementation of edge detection and vectorisation algorithms within this project is the work of the author.

Axiomatic Scan3D is used to produce the sensor scan files and to calibrate the scan coordinate system. These are stored in gCode format [119] which is a commonly used format that can be used to provide machine movement instructions as point coordinates.

Digital image capture was performed using the software provided with the webcam at 640x480 pixels. The digital camera obviously has its own internal memory card for image storage and these images were transferred to the PC via the USB interface.

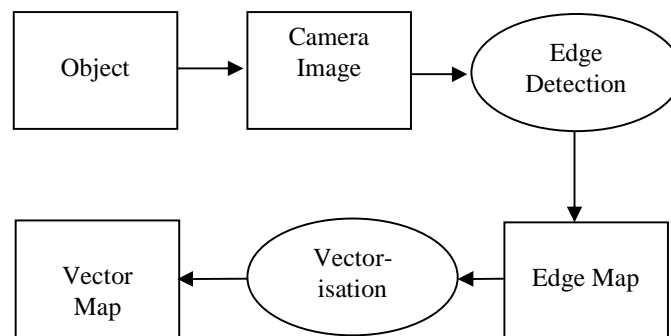
Digital camera images were limited to 640x480 pixels to provide a ‘fair’ comparison with the webcam. Images were stored as standard Windows bitmaps as this format is non-lossy (unlike for example, the commonly used JPG format) and a lossy format could compromise edge detection.

### 5.1.3 Software Overview

The software developed for this project was to fulfil the requirements of producing a scan path / sensor orientations map from a camera image of an object to be scanned. The process of generating the scan path may be split into 2 components:

- 1) Image Processing – this does not require any scale positional calibration. This part of the process is covered in this chapter.
- 2) Path Generation – this requires further input to the system (in the form of calibration data) to determine the position and scale of the object.

A simple overview of the image processing stage is shown in Figure 5-1.



**Figure 5-1: Overview of Proposed Software Design for the Image Processing Stage**

The object is photographed using the selected digital imaging device. The image is then loaded into the test software. Any operations prior to the edge detection such as compensation for camera distortions, or modifications to the image brightness (e.g. histogram equalisation) should be applied at this stage. The operator sets the required edge detection parameters to produce the edge map. The operator then chooses the

vectorisation method and sets the necessary parameters. The vectorisation algorithm provides the data required to construct the scan regions, but to build them with relevance to the original object requires further information about the object's size and location relative to the image processing output. This part of the process is covered in Chapter 6.

#### **5.1.4 Edge Detection Algorithms**

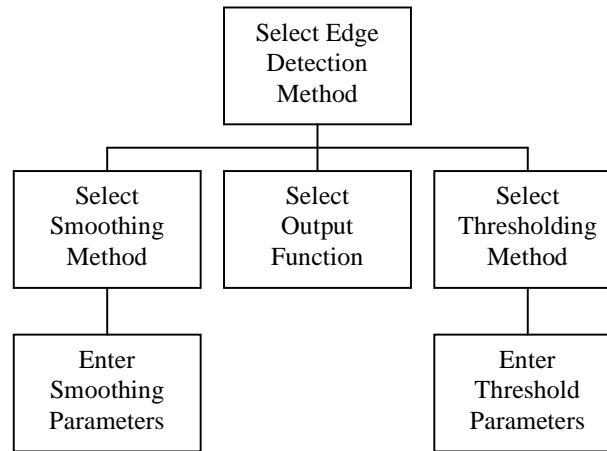
A number of different edge detection algorithms were implemented using a common generic convolution filter designed by the author that allows kernels of any size, number of masks and output function to be applied as a set to an image. Any (first or second derivative of the gradient) edge detector can thus be tested by 'plugging in' the required kernels, thereby ensuring that all operators are handled in the same way as far as possible.

The Roberts Cross, Sobel, Prewitt gradient method, Kirsch and Prewitt compass operators and the Frei-Chen method were implemented as examples of first derivative operators. The Laplacian, Laplacian of Gaussian and Difference of Gaussian operators were implemented as examples of second derivative operators. An implementation of the Canny edge detector was developed, employing a gaussian smoothing function and simple 1<sup>st</sup> derivative masks for detection. The technique employs both the gradient magnitude and gradient orientation map to determine the direction for the edge-tracking and to assign each edge pixel to an orientation group. The theory behind these operators is discussed in Chapter 3.

These operators have been evaluated on a variety of images to assess their effectiveness in edge detection with regard to their suitability for determining the location and orientation of 'geometric' edges in order to provide the basis for developing scan regions based on the discovery of these edges. Section 5.1.5 covers the evaluation of these detectors and provides a number of example output images.

An option is made available to overlay the detected edges over the image in order to observe the accuracy of the edge detection with respect to the original image. If this is not selected then the edge image will be displayed.





**Figure 5-2: Operator interaction required in selecting edge detector parameters**

All the different detectors are handled the same way as far as possible using a common code base. The only differences are in the selection of the masks and the type of smoothing and threshold required by the operator.

#### **5.1.4.1 Convolution Operation**

The general convolution algorithm was designed to only require a single pass over the input image, as the output for all masks is calculated for each point within the image at the same time. Any image positions that are not within the bounds of all masks in the kernel set are ignored.

The convolution algorithm, as developed by the Author, is also used to implement the output for the Gaussian or mean smoothing operation where appropriate as a precursor step to the edge detection. The smoothing and edge detection masks could in fact be convolved together and applied as a single mask (because convolution operations are associative).

Combining the smoothing and edge detection masks into a single convolution mask would further improve the speed of the edge detection process when smoothing is required, but for test purposes these were left as separate stages in order to better judge the results of each operation.

The output function used to determine the resulting pixel value for each image position is applied once all the convolutions for the current pixel position have been applied. The output function is usually defined by the chosen edge detection method, but the implementation allows for selection of alternative methods where appropriate (for example, the gradient magnitude output for the Sobel and Prewitt operators is given by  $G_{out} = \sqrt{G_x^2 + G_y^2}$  however the option to use  $G_{out} = |G_x| + |G_y|$  is also provided as this is faster to calculate).

Determine maximum dimension size for all masks in the kernel set

For each pixel location in the current image row, y

    Access  $y^{th}$  row of image

    For each pixel location in the current image column, x

        For each mask, i, in the set of kernel masks

            For each pixel row of the image, j under the  $i^{th}$  mask

                Initialise result of this convolution  $P_x = 0$

                    For each pixel column of image, k under  $i^{th}$  mask

                        Sum the result of the convolution of the  $j, (k+x)$  image position with corresponding position of the  $i^{th}$  mask

                    Scale  $i^{th}$  result according to 'weight' of kernel mask i

            Increment j

        Store result for  $i^{th}$  mask at position (j, k+x)

    Increment i

    Apply selected output function to all results 0..i for image position x,y

    Store result of output function in output image at x,y

Increment x

Increment y

#### 5.1.4.2 Non Maximal Suppression and Edge Following

The selection of the method of thresholding is also available to the operator where appropriate to the detector. A simple ‘cut-off’ binary threshold level or a non-maximal suppression edge following algorithm (that also thins the edges to a single pixel width) have been implemented. The ‘high’ and ‘low’ threshold parameters are entered by the operator as a percentage of the normalised gradient intensity level.

The edge-tracking and hysteresis algorithm used was developed by the Author, based on the Canny method and employs a recursive process that provides an efficient and accurate edge following algorithm. This algorithm was implemented independently to the prior stages of the Canny detection and can be applied to any of the first-derivative edge detectors (however, it does not work well with the Robert’s Cross algorithm as the calculation of gradient orientation for this operator is not very accurate because of the small kernel size).

The edge following (non-maximal suppression) algorithm requires both the magnitude *and* orientation information for the image. Each pixel is considered and, if it is above the minimum magnitude threshold level, its orientation is considered in order to determine the direction of the gradient for the purpose of non-maximal suppression and which of the adjacent pixels are perpendicular to the gradient for the purpose of edge-following.

If this pixel magnitude value is less than the magnitude of the pixels on either side then it is ‘non maximal’ and will be ignored (i.e. set to zero in the output map). If this pixel magnitude is a maximal edge magnitude and above the high threshold then it is set to an edge.

If it is maximal but between the high and low threshold values then the algorithm must determine if it is linked to a pixel above the high threshold by a chain of pixels (which may also be of ‘intermediate’ magnitude).

In order to follow the edge the orientation is used to determine the direction perpendicular to the gradient and the edge is followed in both directions until either an edge pixel above the high threshold value is found (in one or other direction along the edge) or both directions report that they are not connected to a pixel above the high threshold, and the pixel is set accordingly. The recursive function allows a whole chain of pixels to be followed and marked accordingly.

The algorithm is summarised as follows:

Calculate Gradient Magnitude and Gradient Orientation images

Until each pixel in the image has been checked and status determined as Edge or Not Edge

Status = Follow Edge (pixel)

Where FollowEdge may be simplified into the following recursive routine:

Stop following and mark Not Edge if current pixel position exceeds image bounds

If the pixel is below lower threshold, Status = Not Edge

Else If the pixel is above upper threshold, Status = Edge

Else

    Determine adjacent pixels in direction of gradient and normal to gradient

    Compare the pixel with those adjacent pixels in direction of gradient (and in opposite direction)

    If current pixel is lower in magnitude than either,  
        mark as 'Not Edge'

    Else

    Status =

        Follow Edge (next pixel 'up' in direction normal to gradient) OR

        Follow Edge (next pixel 'down' in direction normal to gradient)

Return Status

Note that extra checks have to be included to ensure that the process does not cycle around a 'chain' of pixels that are between the minimum and maximum threshold levels without being connected to either a definite 'edge' or 'not edge' pixel.

Determining the adjacent pixels depends on the angle associated with the current pixel being examined. The outer loop ensures that each pixel is explicitly visited at least once.

### **5.1.5 Evaluation of Image Processing Algorithms**

Despite many years of research by many authors in the field of edge detection the algorithms are still far from perfect in most general image contexts. Detectors miss often true edges, detect false edges and cause unsatisfactory artefacts of the process (such as edge delocalisation). The edge detection techniques tested show that the user's choice of parameters values in setting up the chosen detection method has a significant impact on the perceived quality of the resulting output.

#### **5.1.5.1 Image Smoothing**

The use of Gaussian smoothing often reduces the number of 'false' edges detected due to image artefacts, and is generally better than a mean average filter of equivalent mask size, however at higher levels of smoothing it causes delocalisation of the detected edges. The optimal level of smoothing depends on the image content, however it is difficult to find a single smoothing level that leads to an optimal detection for all edges in an image. There is always a trade off between the suppression of edges and the delocalisation effect. Another drawback of the Gaussian smoothing algorithm is the width of the mask for larger values of  $\sigma$  (the standard deviation) results in some loss of image data around the image border. However, for this project the position of the object and camera can be controlled to avoid this problem.

#### **5.1.5.2 First Derivative Gradient Detectors**

The Roberts Cross operator is very quick to compute as only four input pixels need to be examined in determining the value of each output pixel, and only subtractions and additions are used in the calculation. Additionally there are no parameters to set. The primary disadvantages are that, since it uses such a small mask, it is very sensitive to noise. It also produces very weak responses to genuine edges unless they are very sharp and determining the edge orientation is not very accurate. There is some ambiguity in the correspondence between the input and output pixels of the Roberts operator, as the operator technically measures the gradient intensity at the point where four pixels meet. This means that the gradient image will be shifted by half a pixel in both  $x$  and  $y$  grid directions. In most practical scenarios the Roberts masks prove unreliable because they are too small to find

edges in the presence of noise. Generally 3x3 masks produce better results than masks of smaller dimensions because they provide averaging of small fluctuations in intensity; however they are slower to compute than the Roberts Cross operator (although with modern PCs this difference is negligible in most situations).

Larger convolution kernels smooth the input image to a greater extent and so make the operator less sensitive to noise. The Sobel operator also generally produces considerably higher output values for similar edges, compared with the Roberts Cross. The Roberts and Sobel gradient masks are more sensitive to diagonal edges. The Prewitt gradient mask is more sensitive to horizontal and vertical edges. The Frei-Chen edge detector has equal sensitivity for diagonal, vertical and horizontal edges.

Most edge detectors work very well on artificial test images (e.g. simple 2D shapes with clearly defined boundaries) however in ‘real world’ images edges are often less well defined and the detected first-derivative gradients in the output image are often several pixels wide due to the width of the kernel operator .

‘Extended’ edge operators are extrapolated from the more usual 3x3 neighbourhood to a similar representation over a larger neighbourhood. To test an extended operator, the Prewitt detector has been extended to use 5x5 masks. The increased spatial resolution serves to reduce the effect of noise or high frequency texture detail at the expense of further ‘thickening’ of edges and increased processing time. The ‘thick’ edges detected can be reduced by using some method of edge-thinning. The non maximal suppression and edge hysteresis algorithm (as employed by the Canny edge detection process) has been used here.

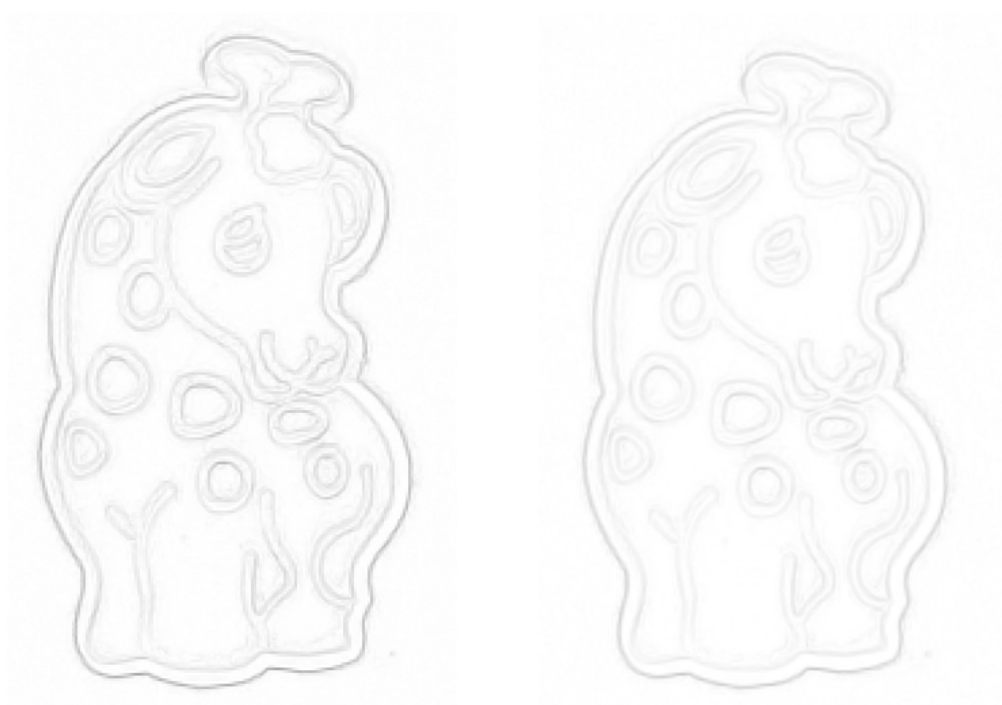
<b>Operator</b>	<b>Mask Size</b>	<b>Number of Masks</b>	<b>Mean time to execute (ms) (av. 5 samples)</b>
Robert's Cross	2x2	2	245.3
Prewitt Gradient	3x3	2	325.5
Sobel	3x3	2	320.6
Frei-Chen	3x3	9	850.5
Extended Prewitt	5x5	2	545.6
Canny (convolution only)	1x3	2	233.5

**Figure 5-3 – table showing comparison of time to execute first derivative operators (all times represent processing of a 640x480 greyscale image)**

The main difference between the results of the first derivative edge detectors is in the strength of the edge response. The extended Prewitt mask might be useful in a situation where there is little contrast in the image and the edge response needs to be enhanced.

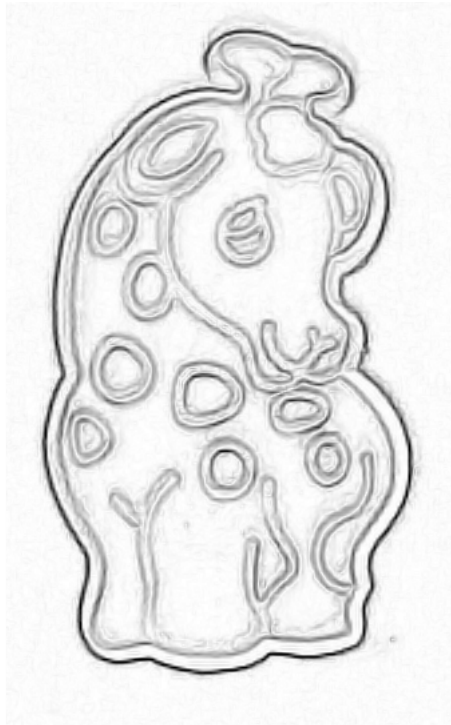
The variation in edge response from the different masks can to some extent be compensated for by scaling the edge output to use the full range of available intensity values prior to further processing (this is done for display purposes within the test program at a later stage).

In conjunction with a thresholding method, such as the edge-following technique used by the Canny operator, most of these edge detectors provide a useful edge response for vectorisation, however the Frei-Chen masks do not specify a means of recovering orientation information so use of this operator was discontinued as the edge-following algorithm requires the gradient orientation for edge-tracking.

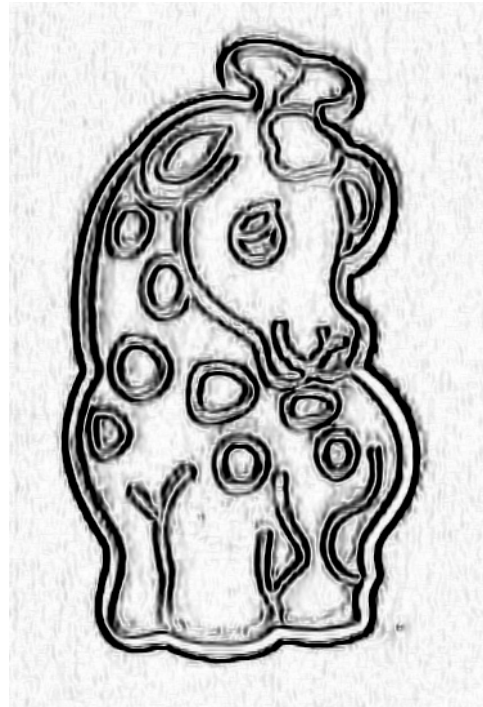


**a) Robert's Cross gives weak edge response. Edge orientation is also poor**

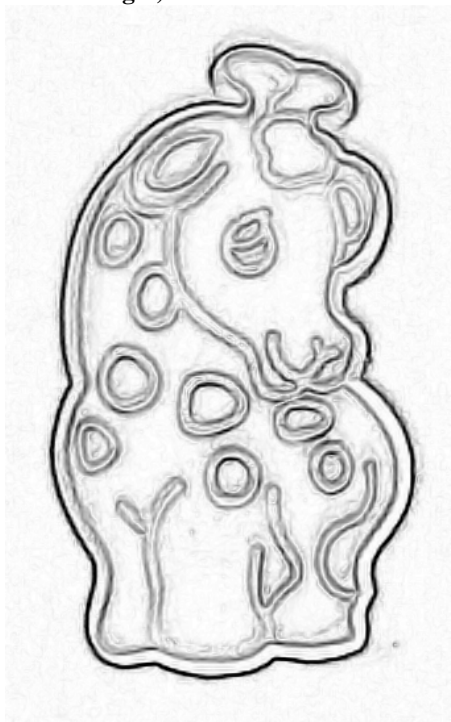
**b) The Frei-Chen edge detector also produces a weak edge response**



**c) Prewitt gradient mask gives stronger response (especially to horizontal and vertical edges)**



**d) Extended. Prewitt Gradient produces a less sharp / more smoothed image, at the expense of stronger 'wider' edge responses**



**e) Sobel operator produces a slightly stronger response than the standard Prewitt mask**



**f) Results of Canny detector (convolution masks only)**

**Figure 5-4 Comparing the output of the first-derivative gradient operators.**



### 5.1.5.3 First Derivative Compass Operators

The Prewitt and Kirsch compass operators provide a means of simple estimation of the gradient orientation rather than a more time consuming calculation based on the magnitudes of the response from the gradient masks, (i.e. the compass edge detection obtains the orientation directly from the mask with the maximum response); however this advantage is undermined by the requirements for 8 convolutions per pixel rather than 2 for the Prewitt (and Sobel) gradient operators.

Figure 5-5 shows the compass operators are much slower in operation than the comparable Prewitt gradient operator – (this is expected, since a set of eight masks must be processed rather than just two). This situation is exacerbated when the NMS / edge-hysteresis thresholding is required because corresponding results are required for both the magnitude and orientation.

Operator	Mean time (in milliseconds) to execute magnitude calculation (N=5)	Mean time (in milliseconds) to execute orientation calculation (N=5)	Mean time (milliseconds) to execute operator +NMS and edge hysteresis thresholding (N=5)
Prewitt gradient (using sum of absolute values)	231.9	237.5	755.2
Prewitt compass	778.7	725.9	1777.3
Kirsch compass	707.2	730.5	1815.6

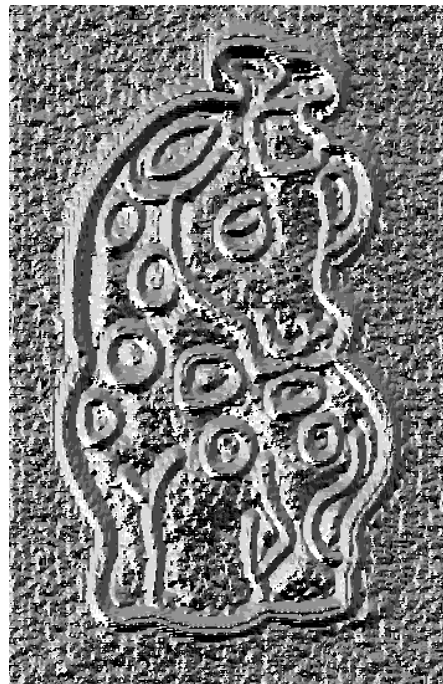
**Figure 5-5: Mean execution time for first derivative gradient compass operators on giraffe mould image (640x480 pixels, 8bpp). (Prewitt gradient operator also shown for reference)**

Figure 5-6 shows the output from the two compass operators tested. A comparative result from the Prewitt gradient detector is also shown. The edge response from either of the compass detectors is not significantly different from that of the gradient detector in these cases, although it can be seen when comparing 5.6d) and 5.6e) that the stronger weights of the Kirsch compass variant emphasises weaker edges than the Prewitt masks (in this case leading to the ‘discovery’ of edges that should be suppressed).

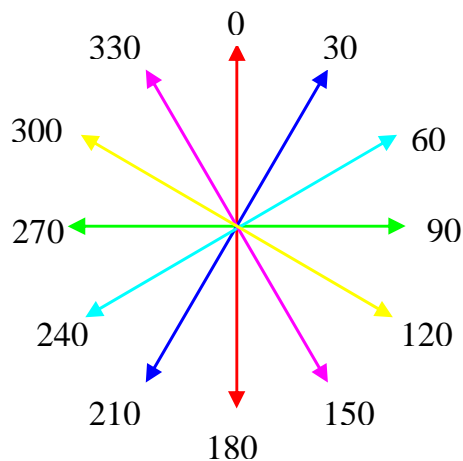
The primary difference between the gradient and compass operators is in the response from the orientation masks. The standard gradient orientation is calculated as a function of the relative magnitudes of the orthogonal mask pair, whereas the compass orientation is selected as a 'maximum' response from one of the mask set. This means that the gradient orientation in the compass image is constrained to be one of the orientations defined by the set (in this case one of the eight rotations of  $45^\circ$ ) whereas the calculation of the orientation for the standard Prewitt operator can be any angle from  $0$ - $360^\circ$



**5-6a) 'Giraffe' cookie cutter mould image**  
(Original  $640 \times 480$  image is cropped to fit).



**5-6b) Kirsch Compass Mask Edge pixel orientation map.** Each  $45^\circ$  orientation is mapped to a greyscale value

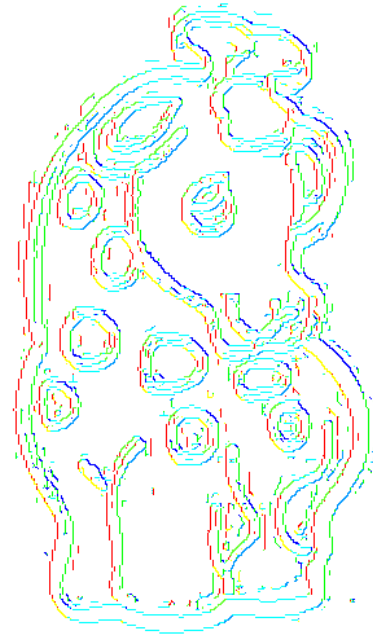


**Colour key to marked edge orientations used in following diagrams.**

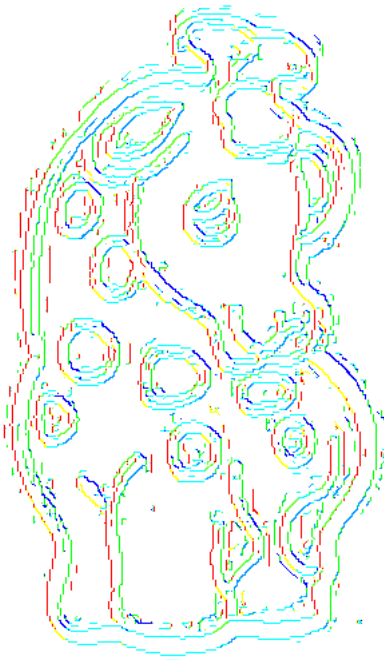
**All edge directions within  $\pm 15$  degrees of the orientation of the edge 'bucket' are assigned the same colour**



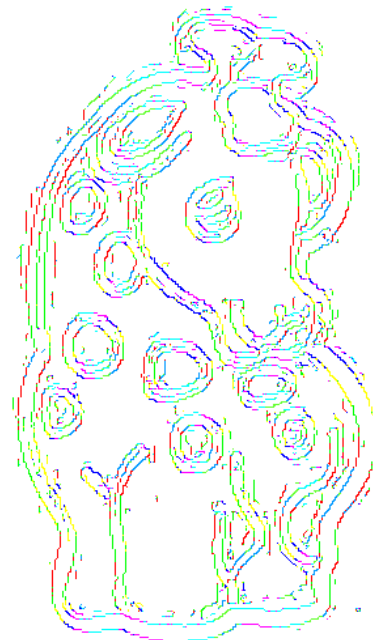
**5-6c) Kirsch Compass Mask pixel gradient magnitude map shows strong response to edges**



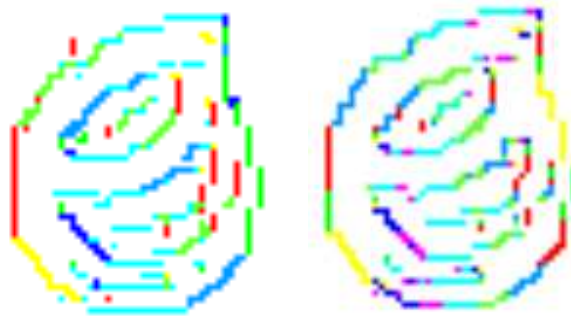
**5-6d) Kirsch Compass Mask combined gradient & orientation map (non max suppression with hysteresis thresholding ( $T_{\min}=29\%$ ,  $T_{\max}=87\%$ ))**



**5-6e) Prewitt Compass Mask with NMS & hysteresis thresholding ( $T_{\min}=29\%$ ,  $T_{\max}=87\%$ )**



**5-6f) Standard Prewitt Gradient operator with NMS & hysteresis thresholding ( $T_{\min}=29\%$ ,  $T_{\max}=87\%$ )**



**5-6g) Close up of ‘eye’ area shows differences between the orientation results for the Prewitt compass (above left) and standard Prewitt (above right). The standard Prewitt kernel allows a more gradual change in orientation whereas the compass orientations are constrained by the masks used.**

**Figure 5-6: Comparing the output from the Prewitt and Kirsch Compass operators with the standard Prewitt kernel.**

### **5.1.5.3 Canny Detector**

The effectiveness of the Canny operator is determined by three parameters - the standard deviation of the Gaussian used in the smoothing phase and the upper and lower thresholds used by the edge-tracking. Increasing the width of the Gaussian kernel reduces sensitivity to noise at the expense of losing some of the finer detail in the image. The localisation error of detected edges also increases as the Gaussian width is increased. The convolution operators are simple 1x3 first derivative masks (as shown in Chapter 3.3.3).

In most cases the upper tracking threshold can be set quite high and the lower threshold quite low with a good tolerance in the results. Setting the lower threshold too high causes noisy edges to break up. Setting the upper threshold too low increases the number of spurious ‘noise’ edges appearing in the output.

The edge following method developed for the Canny operator by the Author is highly efficient at discovering and tracking the lines detected by the first derivative edge detectors. The method of implementation means that it is possible to use this technique with any of

the first derivative edge detectors that have been tested (as long as the edge orientation information is available).

The edge following algorithm (as currently developed) can only follow one direction if it encounters a junction in the edge map<sup>i</sup>. (Such junctions can occur where an edge is partially occluded by another object). In this case the non-maximal suppression step will cause the weaker pixel(s) to be removed and the edge following algorithm will follow the stronger edge. The edge tracker considers two of the ridges as a single line segment, and the third one as a line that approaches, but does not quite connect to, that line segment. This leads to small gaps at junctions where the lines should actually meet and may also lead to edge ‘spur lines’ which in some cases may be a ‘false lead’ (i.e. the ‘stronger’ edge is actually an artefact and the ‘weaker’ pixel follows the true edge direction). Further work is needed into a method of ‘reinstating’ suppressed pixels and possible methods of following lines where they split at junctions.



**5-7a) Results from Canny convolution masks, (no smoothing or thresholding applied).**



**5-7b) Results from Canny operator with Gaussian smoothing sigma 0.75, NMS thresholds  $T_{\min}$  30%  $T_{\max}$  70%**



**5-7 c) As left with edge orientation colour information overlaid on the detected edges.**

<sup>i</sup> This is a common failing of the non-maximal suppression and edge hysteresis technique in general rather than a specific limitation of this implementation.





5-7d) As previous images, with edges overlaid on original image. NB false positive 'edge' detected on shadows along right side



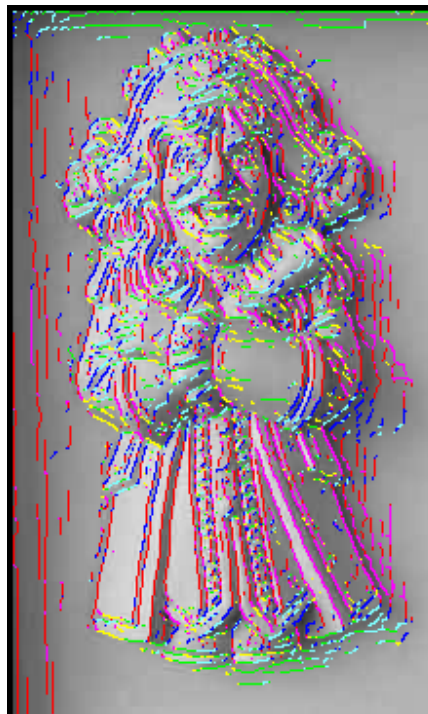
5-7e) Increased smoothing (sigma 1.2) reduces detection of 'fine detail' as well as suppressing some false edges



5-7 f) Higher levels of smoothing (sigma 2) cause further loss of detail and delocalisation of detected edges.



5-7g) Setting THi too high causes only the strongest edges to be detected (30 /95) G1



5-7h) Setting THi too low allows detection of many false positive edges (20/60)



5-7i) Good selection of THi allows TLow to be set quite low without spurious line detection (10/75)

### 5.1.5.3 Second Derivative Detectors

The Laplacian, Laplacian of Gaussian and Difference of Gaussian were implemented and tested as examples of second derivative detectors.

Thresholding of second derivative operators is accomplished by locating the position in the edge image where the zero level is crossed. The zero crossing point corresponds to the position of the edges. i.e. the variation of  $\sigma$  does not affect the location of the zero crossings so localisation remains good. Furthermore, this removes the requirement for the operator to set threshold parameters.

The advantage of this approach compared to classical edge operators of small size is that a larger area surrounding the current pixel is taken into account; the influence of more distant points decreases according to the  $\sigma$  of the Gaussian. Only the position of maximum change in gradient is detected.

Second derivative detectors do not provide any information about edge orientation and the zero-crossings form closed-loop ‘contours’ (except where the edge extends beyond the image area) – this leads to what is commonly called the ‘plate of spaghetti’ effect where the confusion of loops detracts from the appearance of detected edges. The determination of the zero crossings of the second derivative significantly added to the time required to produce the edge output.

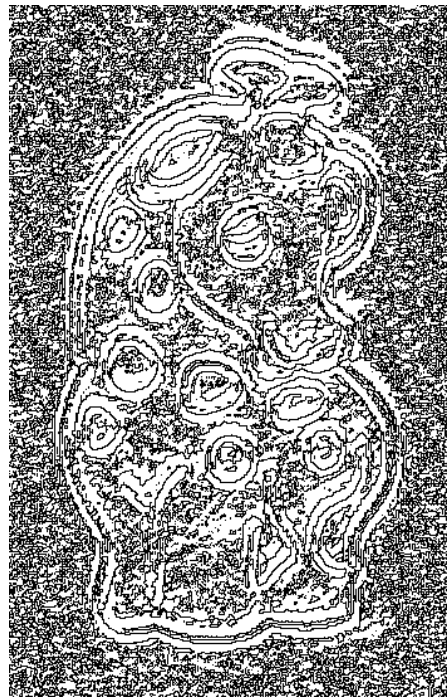
Second derivative detectors are also highly susceptible to noise, particularly where the standard deviation of the Gaussian smoothing function is small. It is common to find many spurious edges apparently detected away from any ‘obvious’ edges. One solution to this is to increase the smoothing of the Gaussian to preserve only strong edges (less significant edges are suppressed), however this has a side-effect of losing corner sharpness, and as the standard deviation ( $\sigma$ ) of the gaussian is increased, the required convolution masks become much larger e.g.  $\sigma = 4$  requires a mask about 40 pixels wide. This increases the processing time and reduces the area of the image that can be correctly processed.

Another approach is to look at the gradient of the LoG at the zero crossing (*i.e.* the third derivative of the original image) and only keep zero crossings where this is above a certain threshold. This will tend to retain only the stronger edges, but again it is sensitive to noise, since the third derivative will greatly amplify any high frequency noise in the image.

In general the results of second derivative detectors appeared to be too sensitive to noise in the image (especially if the image quality was poor). The quality of the results compare poorly to the results from the first derivative detectors. Additionally, the lack of orientation information appeared to be a drawback for the purposes of this project and so the use of second derivative detectors was discontinued, although the requirement to determine orientation information did lead to the investigation and development of the vectorisation methods discussed in Section 5.2



**5-8a) Laplacian edge detection on giraffe cookie cutter mould**

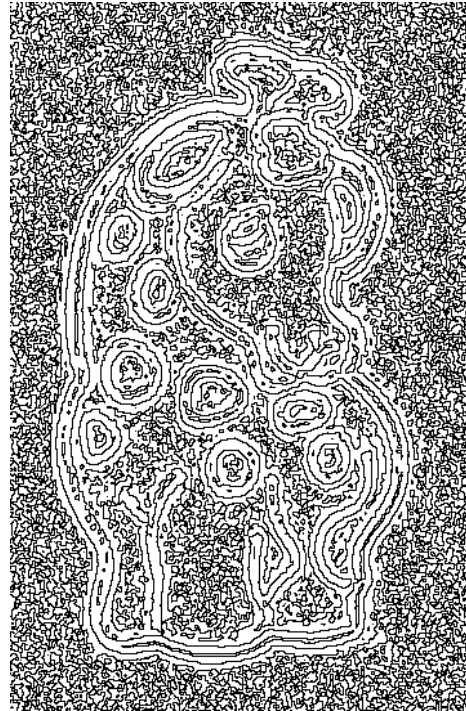


**5-8 b) Zero crossings of Laplacian showing the relatively high level of noise in the detection**





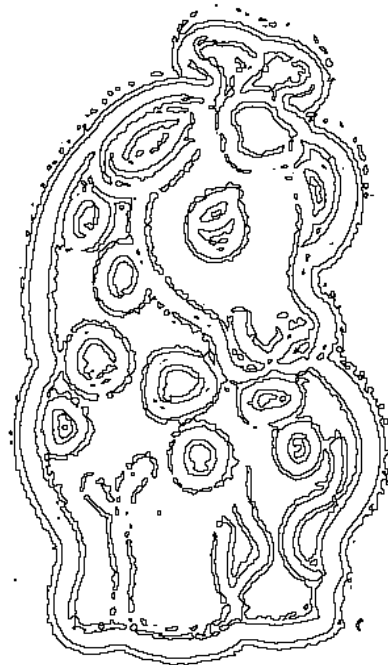
5-8 c) Difference of Gaussian (1:1.4)



5-8 d) Zero crossings of Difference of Gaussian (left) shows slight reduction in noise and slightly clearer definition of 'true' edges



5.8e) Laplacian of Gaussian ( $\sigma=1.4$ )



5.8f) Zero Crossings of Laplacian of Gaussian (left) shows much reduction in the noise level.

## **5.2 Vectorisation Algorithms**

Initially it was hoped that, because the orientation of a gradient can be determined for any pixel that is recognised as an edge, it would be possible to develop scan regions based on the edge pixel orientations directly from the edge detection. Unfortunately this proved unreliable in many cases because the individual pixels sometimes display a local gradient orientation that appears different to the overall trend for an edge as we (as humans) would perceive it to be. This is often the case where an edge has a very shallow (or steep) gradient and the edge must at some point ‘jump’ between pixel rows or columns. Using one of the Compass operators suppressed these fluctuations but limited the selection of edge orientation to the nearest  $45^\circ$ , which was deemed insufficient for the intended purpose of orienting the laser close enough to parallel with the edge to prevent the recognised scan errors from occurring.

Vectorisation of the output of edge detection methods is required in order to determine the trend of edge orientations on a higher level than individual pixels. Two approaches were tested: the Hough transform and a vectorisation method employing a line-fitting algorithm developed in this work.

### **5.2.1 Hough Transform**

The use of the Hough Transform in this project is a means to providing extra information for the extraction of edges. The current implementation is very successful at locating edges in the image; however the visualisation of the edges is unsatisfactory due to the primitive thresholding of ‘significant’ edges. The main problem with the large number of pixels used to provide evidence of lines is the clustering causes the well-known ‘bow tie’ or ‘butterfly’ effect associated with the Hough transform. Only the lines with the very strongest evidence can be separated from the mass of potential lines without this effect becoming prevalent in the image.

Another problem is that evidence for ‘true’ edges located near the periphery of the image that intersect only a small number of pixels within the image space are overwhelmed by evidence for edges that intersect a large number of pixels simply because they occur in the centre of the image. A further problem encountered with the Hough transform is the lack

of information regarding the line endpoints within the image, although this could potentially be addressed by storing the points of contributing pixels.

An algorithm was developed to reduce the bow-tie effect whilst allowing equally valid lines for which only weak evidence exists to pass the thresholding level. This was achieved by iterating a small transform window over the image and performing the Hough transform within that window: the threshold level is therefore applied locally to that window, allowing lines that would otherwise be suppressed. This has the added benefit of localising the Hough line endpoints to the edges of that windowed area.

The Hough transform provides a useful tool in determining where edge pixels may be 'grouped' into lines that represent significant features (and also as a means of eliminating 'noise' or insignificant edges) but further development of this method is necessary to provide a result that is satisfactory for the purpose required by this project. However selecting a correct window size for the image was difficult and a further refinement to the technique was introduced whereby a number of different sized Hough windows were applied recursively and only the edges present at all scales were stored.

This 'recursive' method showed some degree of success in localising the lines discovered by the Hough transform to the edgels from the edge detection image, however the process is even more computationally expensive and often required several minutes to execute for even small test images. There was an effective minimum scale given by a window size of approximately 8x8 pixels, below which the Hough cell accumulator tended to break down due to lack of conclusive evidence for any particular orientation.

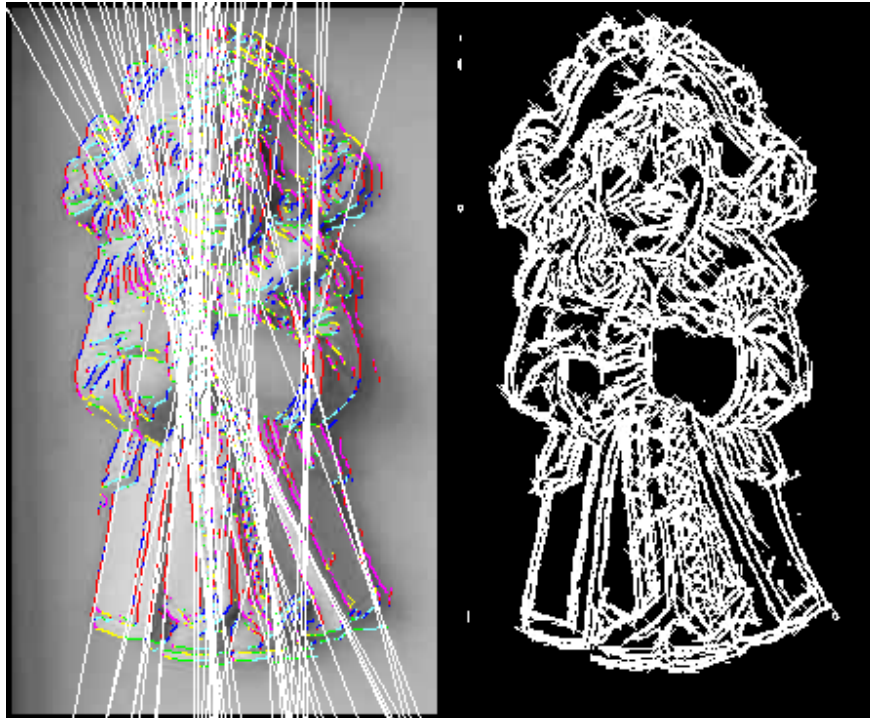


Figure 5-9 shows how the Windowed Hough transform (right) can be used to localise the edge vectors compared to the standard linear Hough transform (left).

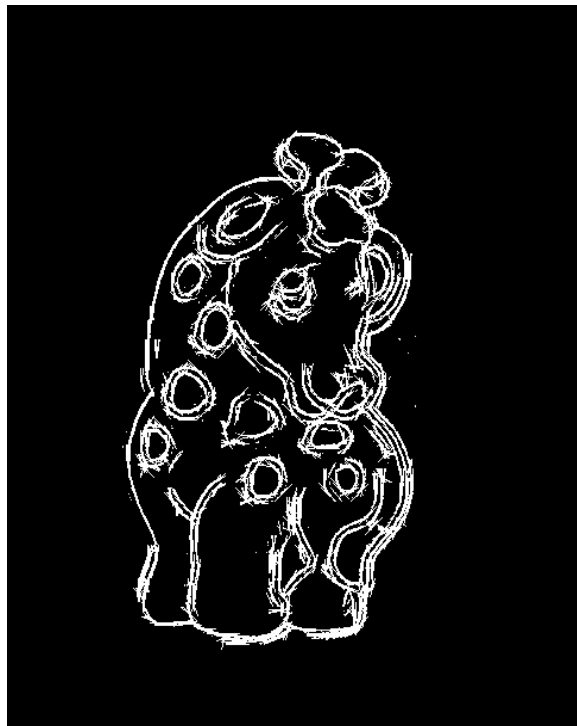


Figure 5-10: Results of Laplacian of Gaussian  $\sigma=1.4$  with 'Recursive Windowed Hough Transform' (window size 12x12 pixels) shows how Hough transform can be used to find the edge orientation from second derivative images.

### 5.2.2 Vectorisation Using Least Squares Algorithm

An alternative algorithm to the Hough Transform method was developed to determine the location of straight line segments in the edge detected image. This approach uses an iterative least-squares line fitting to test the potential vector as the line is grown pixel-by-pixel.

The idea is to test each edge pixel in the image in turn and attempt to ‘grow’ a line starting from that pixel. All eight neighbouring pixels are examined in turn and when an edge pixel is found the ‘growth’ can begin. The line of ‘best fit’ (defined by a start and end pixel) is grown, together with an accompanying list of pixels which have contributed to the development of the line. Line-growing continues until a stop condition is reached. Once the stop condition occurs, if the line has been grown to a sufficient minimum length (as defined by the operator) then it is added to a list of constructed lines.

The first step is to seek to extend the line by looking for an edge pixel, firstly at the pixel closest in orientation to the straight extension of the line and adjacent to the end pixel of the line, and then at the neighbouring pixels on either side (relative to the current line endpoint) if a pixel is not found at the first position. If an edge pixel is found at either of the ‘neighbour’ positions, the pixel closest to the line is added to the list of pixels and a new line is fitted by least squares fitting. Then the signed distances of all the pixels in the list from the new line are calculated.

The ‘best fit’ vector is compared to the edge pixels that are the ‘building blocks’ of that vector and the line must remain within a defined tolerance (in terms of pixel distance and orientation) of the pixels that are used to construct the vector.

The start and end pixels of the new line are chosen as follows. If the magnitude of the gradient of the fitted line is less than 1, the  $y$  value of the start pixel (respectively end pixel) for the new line is changed, if necessary, so that it lies on the fitted line, and otherwise the  $x$  value is changed.

The line growing is stopped if one (or more) of the following cases holds:

1. No extension is possible; because there are no further (adjoining) edge pixels in the direction of line growth. i.e. a genuine 'end of the line' condition
2. The new pixel is already in the list (of pixels contributing to this line). This condition is required to prevent 'oscillation' between edge pixels before the line orientation becomes established
3. The new line has a different start pixel from the original line. The starting pixel is considered to be the 'anchor' for this line and if the starting point of the new 'best fit' line differs from the original starting pixel then the line growing is terminated.
4. The change in direction of the new line compared to the line determined by the previous iteration exceeds a given tolerance: the tangent of the tolerance angle is inversely proportional to the length of the line (so the tolerance angle is reduced as the line grows).
5. The *individual* (perpendicular) distance of one or more of the list of pixels (contributing to the line) from the new line exceeds a given tolerance.
6. The *sum* of the signed distances of all the pixels (contributing to the line) exceeds a given tolerance.
7. The line growth becomes increasingly 'one-sided' (i.e. there is a sequence of pixels in the list with the following property: the first is at a distance greater in magnitude than the mean distance, all are on one side of the fitted line and the number at greater distance than the previous one exceeds a given tolerance). This is typical of an arc.
8. The growth of the line overlaps with an existing (i.e. previously constructed) line. Intersections are permitted, but if the growing line falls within a defined 'angle tolerance' of an existing line then it is terminated. The mean distance is calculated as the sum of the signed distances of all the pixels from the fitted line.

It was found that the above algorithm sometimes produced lines which deviate considerably from the previous trend with the addition of the last few pixels, with the effect of moving the line direction too far in small increments. This situation is not prevented by conditions 5, 6 and 7, unless the tolerances are made smaller to prevent this. However, with the smaller tolerances many lines are terminated prematurely.

Therefore when the growth is stopped under these conditions, an attempt is made to 'rewind' the line until the magnitude of the distance for the last pixel is less than the magnitude of the mean distance for the rewound line. This is done, rather than merely testing for the above condition at each stage of growth, because sometimes the line direction can 'return' closer to the trend after a small deviation.

Values for minimum edge length, vector angle tolerance and an allowance to bridge gaps that may develop in the vectorisation process are provided. The results of the vectorisation process are somewhat dependent on the initial parameters provided by the operator; however the parameters are reasonably intuitive and effective over a range of values. The technique is currently sensitive to changes in parameter values, with small variations producing apparently different outcomes in terms of the actual set of vectors produced, although the different vector sets are fairly similar in their overall coverage in terms of matching the original edges unless the parameters given are largely different.

Currently the distance measurements are given in terms of pixels (as the vectorisation is done prior to the scan region development so is not dependent on 'real world' units of measurement however it is possible, if calibration information was entered, these parameters could be provided in millimeters, which may help further with the operator's perception of the vector image in terms of eliminating / parsing out those vectors which are not representative of edges at a useful scale for building scan regions.

The vectors created are not always 'perfect' matches for the edges discovered in the edge detection stage, however they are generally adequate for the purpose of this project, in that they are accurate and the time taken to calculate the position of the vectors is generally faster than the Windowed Hough method, although it can still take some minutes to determine all the vectors in a complicated image such as the giraffe mould.

### 5.2.2.1 Vector Reduction

In order to keep the vectors fitted closely to the edges discovered in the image the initial vectorisation tolerance parameters must be quite strict. Simple (artificial) images reduced to a minimum sufficient set of vectors quite efficiently using this method, however in ‘real’ images there is often some redundancy in vectors, given that the edges are often curved and a number of straight lines may be fitted around that curve. This often leads to the generation of many short line segments. Such vectors usually overlap along part of their lengths and if the lines are within a given angle tolerance they can be merged to create a single, longer vector.

Another common situation occurs as an artifact of edge detection whereby a double edge, separated by a gap of one or two pixels, occurs due to light catching an edge and causing a peak in image intensity along the edge. It may be beneficial in these circumstances to consider close, parallel edges as a single edge vector and therefore to merge them.

Also there are situations where discovered vectors may be merged, where the initial tolerances may have been too strict causing the break-up of long vectors into shorter segments. The vectorisation algorithm may also result in the generation of parallel overlapping segments from a single line. In such cases it is possible to merge these segments into a single vector using a ‘combine’ algorithm.

The merge process is non-trivial as the position of the ‘combined’ resultant vector position may be shifted from the position of either of the contributors and the algorithm must consider the relative lengths of the pair of vectors concerned as a weighting.



### **5.3 Summary**

In this chapter the results of the implementation of a number of common edge detection methods on a common software platform have been discussed with respect to their accuracy and timing requirements. It was found that, given the CPU speed and available memory of current PCs, all the edge detection algorithms tested performed well in terms of speed, and where the non-maximal suppression and edge following algorithm was applied with acceptable parameters provided by the operator, most of the algorithms provided a usable edge image. The implementation of the detectors as 3 discrete stages allowed for any chosen combination of smoothing and convolution filter and thresholding method to be used. It was found that there was little difference in the effective results of the first derivative edge detectors tested, with the exception of the Roberts Cross which was deemed too weak in terms of edge response and gave poor results in terms of edge orientation.

In almost all cases it was found that the Canny algorithm, with suitable parameters provided by the operator, produced results that were deemed acceptable. As such all work with respect to vectorisation and scan region development was based on the use of the Canny filter with suitable smoothing and threshold parameters to be provided by the operator on a per-image basis.

It was initially hoped that the edge detection image could be used directly in the development of scan regions for the direction of the laser scanner however, because the calculation of edge orientation within the edge detection process is very localised (i.e. calculation based on a per-pixel basis), the indicated orientation for certain pixels could sometimes differ considerably from the orientation that a human operator would perceive as the line to which that pixel contributed. As this became apparent, methods by which the discovered edge pixels could be accumulated to provide information at a 'higher level' were explored.

The standard linear Hough Transform was investigated initially, however this was found to provide vectors which did not have known endpoints and as such extended all the way across the image area. Attempts to develop an algorithm by which the vector endpoints could be localised through a 'recursive windowed Hough transform' method were found to be extremely costly in terms of processing time and memory resources to provide any

useful level of edge localisation and a new approach was developed by the Author using a 'least mean squares' line fitting algorithm which proved to be reasonably effective and relatively quick to execute compared to the windowed Hough method for the images under consideration. It was decided to use this method due to time constraints on the investigation of further vectorisation methods such as orthogonal zig-zag [120], skeletonisation or medial axis transformation [121] of thresholded images, however these methods were not tested here as the non-maximal suppression algorithm already provides edges of single-pixel width.

Other image processing methods for determining scan orientation regions, such as region and texture detection are considered as enhancements in Chapter 8 as possible directions for future work.

---

<sup>116</sup> Ruzon M. & Tomasi, C. '*Color Edge Detection with the Compass Operator*', In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Ft. Collins, CO, V. 2, pp. 160-166, June 1999

<sup>117</sup> Novak, C.L. & Shafer, S.A. '*Color Edge Detection*', Proceedings DARPA Image Understanding Workshop, Vol. I, pp. 35-37, Los Angeles, CA, USA, February 1987.

<sup>118</sup> [www.paintlib.de](http://www.paintlib.de). Last accessed 5/9/06

<sup>119</sup> National Center for Manufacturing Sciences, '*The Next Generation Controller Part Programming Functional Specification (RS-274/NGC)*', Draft; NCMS; August 1994

<sup>120</sup> I. Chai and D. Dori, '*Orthogonal Zig-Zag: An Efficient Method for Extracting Lines From Engineering Drawings*,' C. Arcelli, L. P. Cordella and G. Saruiti di Baja, eds., Visual Form. New York: Plenum Press, pp. 127-136, 1992,

<sup>121</sup> <http://homepages.inf.ed.ac.uk/rbf/HIPR2/skeleton.htm>

## 6. Data Fusion Algorithms

The use of image analysis as input from one sensor as a guide to determining the most correct orientation of another type of sensor in order to improve the performance may be regarded as ‘sensor fusion’ i.e. data from one type of sensor is used to select data from another type of sensor.

The conceptual outline of how the location of detected edges may be used to provide a set of partial scan regions that together provide a complete scan of the object is provided in Chapter 4. Chapter 5 shows how the image analysis can be used to provide a vector map of discovered edges in the camera image. In this chapter it will be shown how the vector map can be used to provide a guide to the laser scanner orientation.

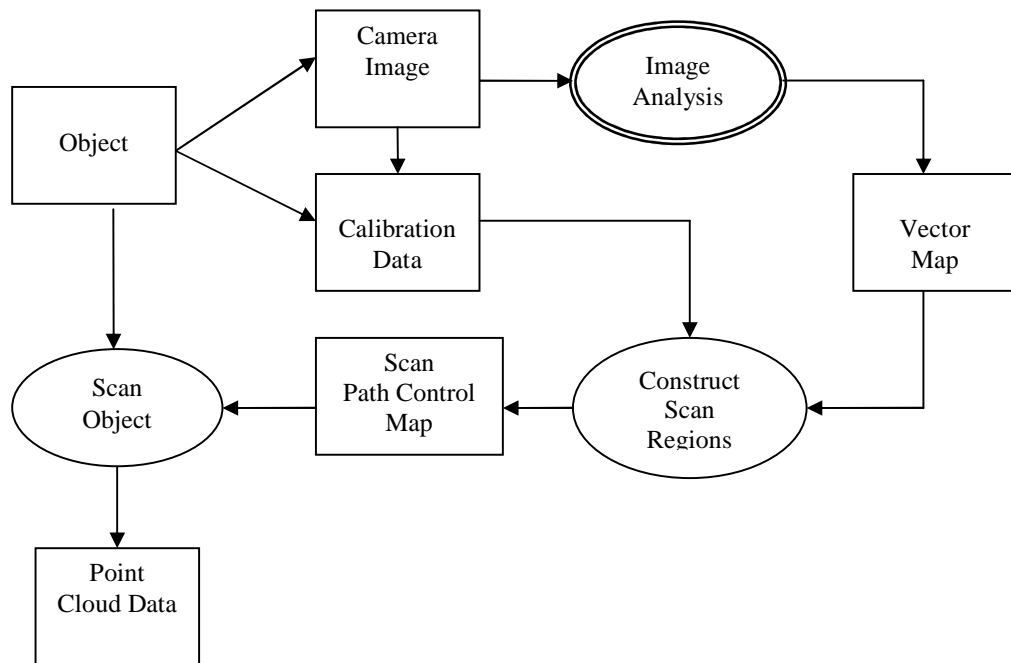
Two main approaches were implemented based on the ideas described in Chapter 4:

- 1) Building scan regions of a pre-defined size around each edge vector.
- 2) Selecting the orientation at each point based on that of the nearest vector.

The performance of both these methods will be compared against the ‘optimal’ single-scan direction method, which is also detailed below. This chapter concentrates on the implementation of these methods and a discussion of some of the limitations imposed by the existing system. The comparison of the results of these different methods on several different objects is given in Chapter 7.

## 6.1 Software Framework

In Chapter 5 the software described was developed to implement and test a number of edge detection and vectorisation algorithms. These image analysis steps result in a ‘vector map’, which is a list of discovered edges in ‘image space’. In order for this map to have relevance to the scan data space it must be calibrated and scan regions constructed according to the algorithms described in Chapter 4. An overview of the software data flow is shown in Figure 6-1. (The edge detection and vectorisation stages described in Chapter 5 are summarised as the ‘Image Analysis’ process in this diagram).



**Figure 6-1: Overview of Software Design Data Flow**

The scan region map will be in the form of machine instructions that indicate the correct orientation of the laser for any position within the scan area. Calibration data is required in order to relate the scan region to the scale of the object in the ‘real world’ coordinate system of the CNC machine. Ideally, calibration points should be extracted from the image automatically, however a manual calibration process is used in the test program.

This information is then read by the scan control program (Scan3D) and used to manage the scan process: all scan regions of the same orientation are scanned, then the laser orientation is changed and the next orientation set is scanned until all regions are covered.

In the case of complex objects where the determination of scan regions may take a few minutes it is possible to establish an initial 'dominant' scan direction in order to begin scanning before all the software calculations are completed. This will provide a 'base' scan onto which the remaining orientation 'partial' scans will be fitted.

Due to hardware limitations it became clear that the CNC machine controller would not be able to support the automatic rotation of the sensor head during the development of this project, nor was the Scan3D application able to allow multiple scans to be combined into a single composite point cloud.

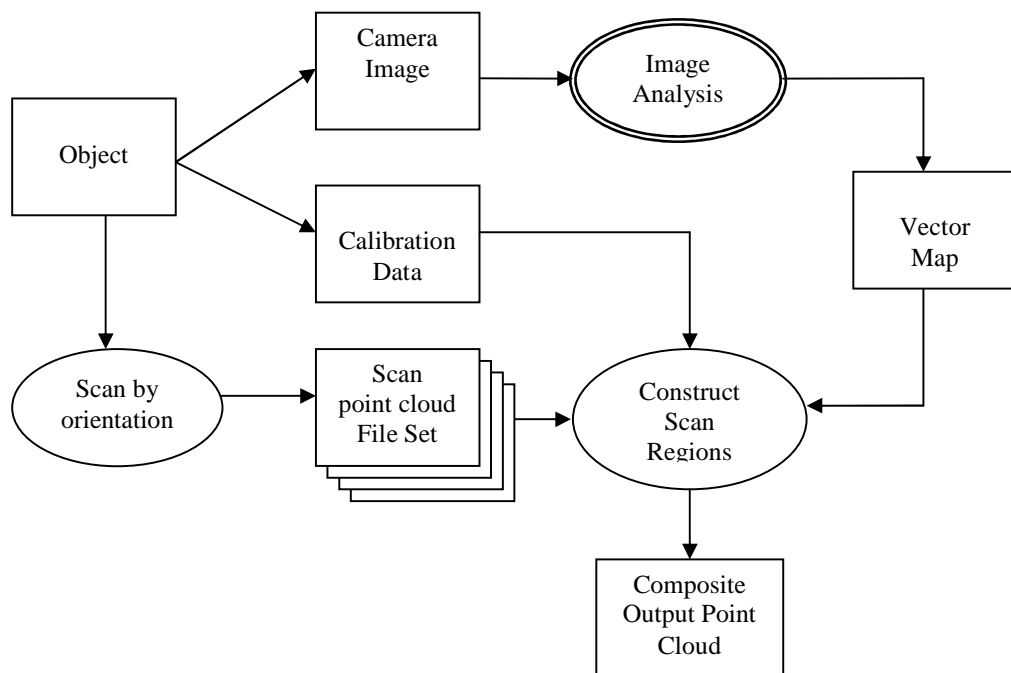
It was hoped that Axiomatic Technology would be able to provide this functionality within the Scan3D application during this project however due to other commercial commitments the development of these features was not completed. Because of these hardware limitations it was necessary to take a different approach in order to provide a simulacrum of the composite output file.

In order to conduct the investigation without needing to perform the scans every time a new strategy was tried. The scan data was collected as a set of complete scans and then virtual partial scans could be performed by using software to select the relevant data from each scan.

Thus, each test object was scanned at a number of chosen orientations equally distributed through a range of 180°. Usually a set of 10 scans were performed at 18° intervals. This separation was chosen to correspond to the earlier experiments where it was shown that errors were minimal if the difference between the true edge orientation and the scan orientation was less than  $\pm 10$  degrees, and facilitated by the fact that the stepper motor used to control the sensor rotation has a rotation of 1.8° per step.

In an ideal system each geometric edge would be scanned at the ‘nearest’ step orientation to the reported edge orientation, thereby reducing any errors that may be seen in the composite test image (although, as suggested in Section 4.1 the grouping of edges into ‘bands of closest orientation’ may be used as a strategy to reduce overall scan time without severely impacting the overall scan quality).

This strategy also allows software testing of changes to the image processing and scan region parameters without requiring that the object is rescanned after every change in order to see the results. As the scanning operation is by far the most time-consuming aspect of the whole process, the ability to re-use a set of scans for a given object is advantageous in testing the effects of changing scan parameters on the output.



**Figure 6-2: Overview of Software Design As Implemented for Testing**

### 6.1.1 The Scan Acquisition Process

For the test program it is necessary that a set of complete scans are performed for each object at known orientations. Before commencement of the scanning process, it is essential to ensure the experimental rig and the mounted sensor are set up and calibrated appropriately.

The complete scanning process for an object is as follows:

- 1) The laser sensor and camera are mounted securely on the CNC machine. Care must be taken to align both sensors as close as possible so that their axes are parallel to each other and perpendicular to the scan bed in order to minimise calibration errors.
- 2) The object is treated with a non-permanent coating of diffusely reflective spray if it is shiny, or of low reflectivity, or has highly contrasting colours. This is to ensure a homogenous surface reflectivity, thereby reducing some of the possible stochastic and systematic distortions.
- 3) A thin cardboard layer of a plain contrasting colour to the object is placed on the scan bed and taped down to provide a flat and clean surface that will minimise false edge detection. The object to be scanned is placed on the card.
- 4) The scan area is defined using the Scan3D software. This covers an area wider than the object so that the effect of applying the scan region algorithms can be observed around the edges of the object. The sensor's initial point is defined as the lower left corner of the scan area. When multiple scans of an object are made, the origin is used for registration because the multiple scans share this common base coordinate.
- 5) Calibration markers are placed around the object at known scan coordinates (e.g. the four corners of the defined scan area can be used) in order to correlate the position of each object in the image space with the scan data. Each marker is placed by using the laser spot as a guide to locate the position for each marker.
- 6) The object is then digitally captured using the webcam or digital camera mounted on the CMM scan head. Two 640x480 greyscale digital images are taken: one with the calibration markers in place and another with the markers removed. It is important that the camera position remains undisturbed for both images in order for the calibration



points to be consistent. The pixel coordinates of the markers in the image space is recorded and used with the known scan coordinate positions to determine the affine transformation for the calibration.

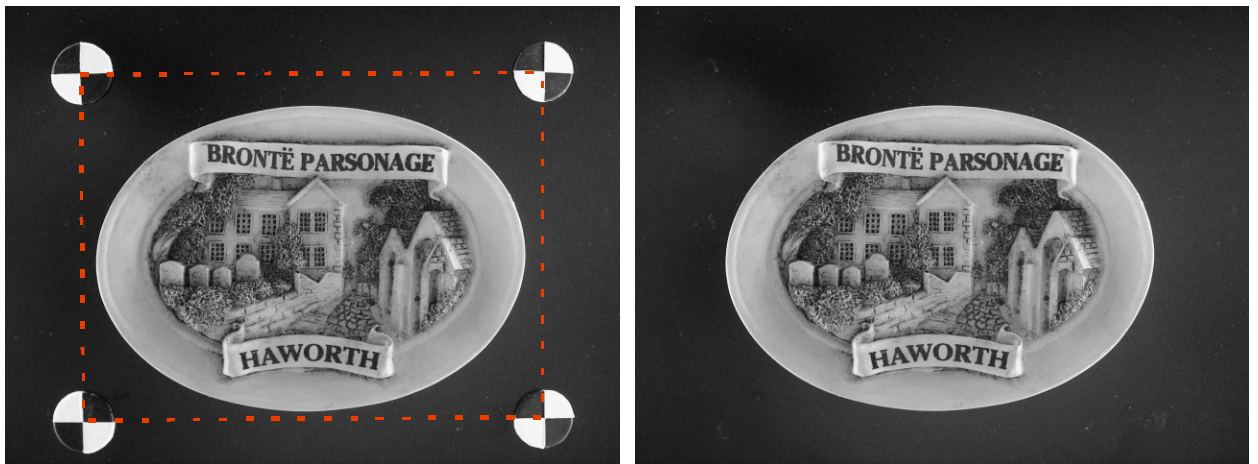
- 7) The Scan3D software is then set up to determine the density of 3-D points to be collected, scan speed, measurable range (highest and lowest scan points), scan direction, and other relevant parameters.
- 8) The scanning process then begins: the sensor traverses the  $x$ -axis of the CNC. The height and orientation of the scanner remains fixed throughout the scan. Each scan line is recorded by the Scan3D software for every complete traversal of the sensor with the chosen orientation. This procedure is repeated for all scan lines until the entire defined scan region has been covered.
- 9) The acquired range image is then saved as raw point cloud data so that post-processing can be performed (i.e. the image analysis can be applied to the set of point cloud files). Scan3D supports a number of commonly used formats. For the purposes of testing the gCode format has been used.
- 10) Multiple range images of an object are required by the test program at a number of orientations. The sensor is set to the next required orientation and its position is adjusted until the spot image is again on the defined origin. The initial parameters of the experimental rig are maintained and steps 7 – 9 are repeated until all the required range images are acquired.

Once all the scans are acquired, they are loaded as a set into the image analysis program and, along with the calibration data, used to produce a composite output data file corresponding to the output map created by the chosen scan region algorithm. This composite file is then exported in a format suitable for display as a 3D graph.

### 6.1.2 Image Scan Calibration

The data set, stored as gCode has a coordinate system provided by the Scan3D program based on the CNC machine coordinates. In these scans the origin is at the bottom left corner of the CNC scan bed and the initial point of the scan is offset in x and y (measured in millimetres) from that position.

The camera image and resulting vector map output coordinates are known only as pixel coordinates relative to the top left corner of the image. For the vector map to be interpreted as data meaningful to the laser scanner some means of relating the two coordinate systems is required. A simple manual calibration system is employed here, using a calibration image which is identical to the digital image used for the analysis process with the addition of calibration markers placed at the four corners of the area covered by the scan set, as shown in Figure 6-3



**Figure 6-3: Corresponding Images for Calibration and Analysis.** The red outline in the left picture shows the extent of the scan area, with the calibration markers places in each corner.

The coordinates of the origin and maximum extents of the scan are stored in the scan file so it is therefore possible to relate the pixel coordinates of the markers to the scan coordinates. The corresponding data points are entered into the image analysis program and an affine transform as described in Section 4.3.3.1 is used to determine the transformation required to relate these points. Once the calibration has been performed the chosen scan region algorithm is applied to generate the scan region map.

## **6.2 Experimental Scan Region Generation Algorithms**

In the experimental version of the software the object was scanned at a number of known orientations and the region orientations were selected to match with the scan files. As such, the implementation of these algorithms is required to work from a slightly different perspective than a ‘real’ implementation. The system limits the precision to which edge orientations may be matched, although the choice of the scan orientations in this system was based on earlier experiments where it was shown that the magnitude of scan errors when the sensor orientation was less than  $9^\circ$  off parallel to the edge were minimised.

It was decided to scan each object at  $18^\circ$  increments of sensor rotation through  $180^\circ$ . (The full  $360^\circ$  rotation required by Wong’s implementation is not required as this method seeks to keep the laser parallel to the edge at all times rather than selecting a correct value from a complete set of ‘known’ scans). This  $18^\circ$  increment was also partly influenced by the step-rotation of the sensor, which has a single-step of  $1.8^\circ$ , so each scan represents a 10-step rotation of the sensor. Any detected edge is therefore no more than plus or minus 9 degrees from a scan orientation file and therefore the correctly-selected value should display minimal errors.

### **6.2.1 Dominant Single Orientation Scan**

For all the possible sensor orientations there will be one scan that generates the smallest number of errors simply because its orientation conflicts with the least (summed) length of edges in the object. If this scan orientation can be determined it provides a simple means of determining a ‘benchmark’ orientation to which the other algorithms can be compared. For the ‘intended’ working system it may also be used to provide an initial scan orientation onto which the partial scan methods may be ‘patched in’.

The determination of this algorithm is relatively quick and simple and therefore it could be used to determine the initial sensor orientation. Then the scanning process using the determined orientation could be started whilst the remaining partial scan regions were calculated using one of the later algorithms. This method also has the advantage that it does not require calibration information in order to provide its output – all the processing occurs within the image analysis domain.

### Algorithm 1 – ‘Dominant Single Scan Orientation’

```
Determine the required number of potential orientations,  $N^i$ 
For all pixels in image
  Determine pixel orientation  $\theta$ 
  Add count for pixel to orientation  $N_\theta$ 

Set max_count to value for  $N_0$ 
Set max_index to 0

For each orientation 1 to N
  If value of  $N_\theta$  greater than max_count
    Set max_count to value at  $N_\theta$ 
    Set  $N_{\max}$  to  $N_\theta$ 

Orientation  $N_{\max}$  is determined to be the orientation corresponding
to the most prevalent orientation in the image
```

In the initial case, this algorithm was applied to the edge detection image, however the edge orientation detection determines the orientation at a local level (i.e. individual pixel) rather than an overall trend in edge direction. This can result in spurious results in real-world images where edgels are not always ‘cleanly’ detected, or where the pixel resolution forces the edge to ‘choose’ between two adjacent pixels. Where a detected edge vacillates in direction and falls close to a boundary between two orientation groups the edge may contribute to both orientation ‘buckets’, thereby weakening the influence of that edge to a single orientation.

Because of these effects the decision was made to vectorise the detected edges. Once the vectorisation algorithm was written Algorithm 1 (above) was modified to sum vector length by orientation (rather than counting raw pixel data) in order to decide the dominant orientation.

An improvement to this algorithm would be to use the maximum number of possible orientations, then apply a ‘sliding window’ over a range of contiguous orientations and count the total number of pixels falling under the window. The number of orientations under the window should be the width of the ‘angle tolerance’. This might give a better result than a single ‘best’ orientation and removes the requirement for the user to decide on the number of fixed orientations (they might still be required to decide the angle orientation tolerance although this could potentially also be automated).

---

<sup>i</sup> in the test program this is equal to the number of files in the scan set

### 6.2.2 Partial Scan by Region Orientation

The theoretical basis for this algorithm is discussed in Section 4.1.3.1. The practical implementation of this concept is considered here. In order to determine whether a certain point falls within one or more scan regions it is first necessary to decide how many orientations are required in the scan. In the test program the number of orientations used corresponds to the number of scan orientation input files.

The orientation of each vector in the vector map can be simply determined and the vector used as a template for the creation of a scan region. Each such region is extended around its parent vector by a distance known to encompass the area that would include any scan errors created by scanning the real edge with the sensor at a perpendicular orientation to that edge vector. (i.e. in an ideal situation a scan region exclusion zone is created within which the only scan orientation employed is parallel to the edge).

It is likely that, in practice, such scan regions will overlap for any real image. Because no orientation is more important than any other, there must be some means of resolving the selected value at those positions for which there is a conflict of orientations. Overlapping regions of the 'same' orientation are not currently counted more than once.

The list of vectors is processed to determine the orientation for each scan region, and thus the layer in the orientation stack to which it should be written. The area representing the scan region for that vector is filled using a simple polygon fill (draw) algorithm, using a value that represents the orientation.

An 'image layer' is created for each scan orientation and for each vector the area covered by its scan region is written to the corresponding orientation image layer using a simple 'polygon fill' algorithm. Therefore a stack of 'exclusive' orientation layers is created containing the only marked areas of the image that must be scanned at that orientation: each layer can then be used in the 'real' program to generate a scan path file that can then be assimilated to produce a scan that covers the whole of the scan area. Overlapping scan regions of the same orientation will be 'merged' whilst those of different orientations will be treated separately. Effectively a stack of images exists, with each layer representing one of a defined set of orientations, for all points in the image. If a line is then drawn vertically through the stack for any given point in the scan space it is possible to transform the point to image coordinates to determine which scan regions are active at that point.

In order that no ‘holes’ are left in the scan by the conversion of scan coordinate points by affine transformation to image points, the orientation layer with the greatest ‘filled’ scan area can be used as a ‘base’ scan orientation (similar to that used for Algorithm 1) onto which the other orientation layers can be patched.

One effect of this method of separating the regions into different orientation layers but merging those regions that fall within each layer is to change the balance of the image with respect to the output ‘base’ scan orientation. As scan regions of similar orientation (i.e. within each layer) are merged some of the ‘weight’ of that orientation is lost (as each pixel represents a scan point that will be scanned once it is only counted once). This presents a more sensible method of calculating the base scan orientation than the other methods, which simply sum the vector length per orientation without concern for the actual scan area covered by that orientation.

## **Algorithm 2 – Partial Scans by Scan Region Orientation**

```

Determine number of required orientationsii ( $N_\theta$ )
Create  $N_\theta$  orientation image layers

For each vector
    Determine its scan region and orientation
    Fill the corresponding area of the indicated orientation layer

Determine primary Scan Orientation based on total scan area per
orientation.

Set output data file to copy input file corresponding to the primary
scan orientation

Use calibration data to determine correspondence between scan regions
image coordinates and actual scan points

For each point in the output scan file
    Look up indicated orientation(s) in orientation image layers

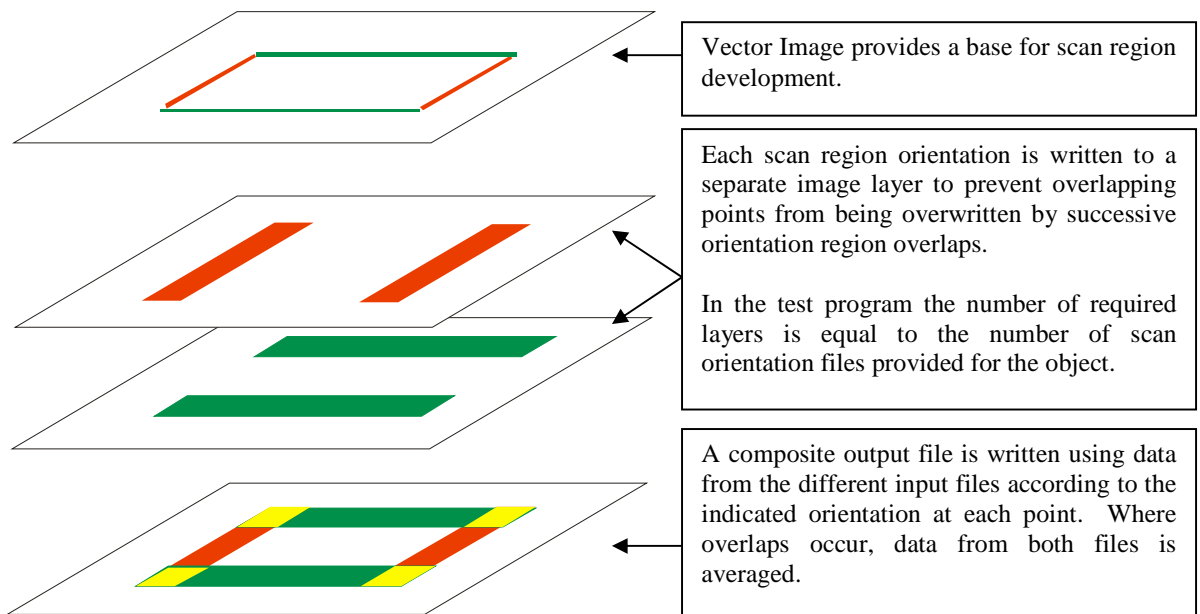
    If orientation is not primary orientation
        Determine single output value from all indicated input
        orientation values

        overwrite point in the output file with new data value

```

---

<sup>ii</sup> For the test software, this is equivalent to the number of scan files. In the real world this would be entered as the required orientation precision.



**Figure 6-4 Shows the concept of the orientation image ‘stack’ whereby scan regions belonging to each defined orientation are written to separate layers to avoid loss of information whilst allowing regions of similar orientation to be merged to reduce redundancy.**

In order to determine which scan regions are active for each point in the scan space the transformed point in the image space is checked for all orientations in the stack. For any layers active at that point, the value from the corresponding scan file at that point is selected. Potential methods for resolving data points within overlapping regions are discussed in the following section.

#### **6.2.2.1 Reconciliation of Data for Overlapping Scan Regions of Different Orientations**

One problem with this method is how region overlaps may be resolved. In many places in the image, but especially at corners (i.e. vector intersections) there are areas in the image covered by more than one scan region. Because the height values for those different orientations are likely to be significantly different it is important that these regions of overlap are resolved if the data is to be useful.

Where regions overlap this indicates that some doubt exists as to which is the ‘correct’ orientation and often therefore there is usually some disagreement between the values from the corresponding scan data files. In cases where the orientations are similar (for example,

a scan region at  $18^\circ$  overlaps a scan region at  $36^\circ$ ) the two input values may be quite similar and the resulting output value may be averaged with no noticeable problems. However there will often be situations where scan regions representing edge vectors of very different orientations (e.g. a  $0^\circ$  scan region overlaps a  $90^\circ$  scan region) where the values are likely to be very different.

In the test system the data from the orientation scan files must be resolved at the corresponding data points where they overlap within the area of indicated scan regions, and there would be a required analogous step in the real system whereby the data from overlapping partial scans would also require a means of determining a single value. The situation becomes even more difficult if there are more orientations overlapping at the same point.

In the experimental version the decision was taken to use a simple average of all the values that are active for a given point. This provides a reasonably smooth transition from one region to another where the orientations do not differ by more than one or two orientation steps, however where there are large differences in the orientations the averaging does not always provide very satisfactory results, as (at least) one of the orientations will yield an incorrect value, often from an area where a spike / trough or bow wave is present. Where multiple scan regions overlap (i.e. more than two different orientations) it may be possible to use a weighted average to discard outlying values, however there is always the possibility that the outlying value may in fact be the most correct one.

Another possible method of determining a single output value may include determining a weighted average of the values within the region of overlap that would involve some blending of the values based on their contribution to the current point as a distance from the longitudinal axis of their respective scan region. Other considerations towards the weighting could include the relative sizes (i.e. influence) of the overlapping scan regions although it could be argued that this is irrelevant as it a small, localised scan region could be providing better data than the large region that it overlaps.

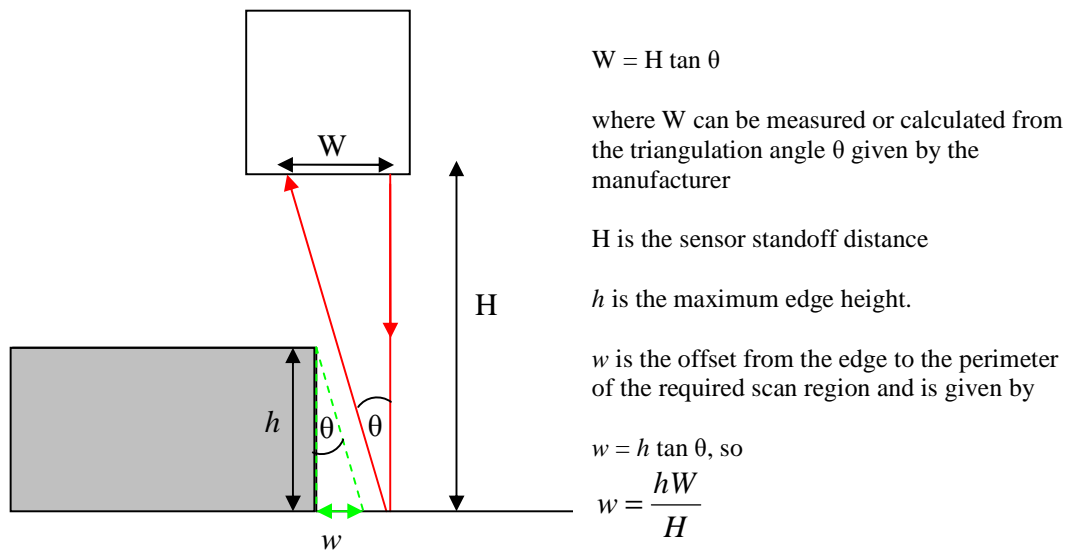


### 6.2.2.2 Width of Scan Regions

The scan region must be wide enough to catch any potential error from scanning at an incorrect orientation relative to the current edge; the scan region is in effect a ‘safe zone’ in which the influence of the vector (around which the scan region is created) extends control.

The width of the scan region required is determined from the scale of the image (from the calibration data) relative to the approximate maximum edge height of the object (which can be measured approximately given the scale of objects commonly used with this type of scanner). The length of the scan regions is also extended beyond the ends of the vector to cover any small gaps that may appear in the edges as a result of the vectorisation process.

The triangulation angle of the sensor employed is known from the manufacturer’s data sheet, and so the width of the scan region required to avoid the occlusion effects may be calculated by the similar triangles method, as for vertical edges, which cause the worst case errors (as previously shown), there is a simple relationship between the triangle described by the path of the reflected laser beam from the emitter to the detector and the maximum occluded distance created by the object’s edges. (See Figure 6-5).



**Figure 6-5: Scan Region Width calculation by similar triangles with respect to the triangle created by the path of the sensor beam**

However the scan region must be extended by this distance on both sides of the vector because, prior to scanning, it is unknown which side of the edge vector represents the ‘high’ side and which is the ‘low’ side (with respect to the object height in the scan space).

The determination of an algorithm to provide the necessary scan region width, in order to cover the full width of the area susceptible to bow waves in a general situation proved troublesome because the width of the bow waves is often specific to the particular combination of surface reflectance of the edges under consideration. The relationship between the reflectivity of the surfaces involved is unknown at this time so was not included in the calculation for region width. (This is covered in more depth in Chapter 4.1.1 on Orientation Tolerance). However, as most geometric edges of the object under consideration are usually less than the full permitted height of the sensor range, it is possible to allow wider scan regions by setting the edge height,  $h$ , to be the maximum permitted object height, in order to catch more of the area of potential secondary reflection errors within the scan region. Unfortunately increasing the scan region width usually has the unwanted side-effect of causing more scan region overlaps and therefore more averaged data output. Also, from the perspective of producing a laser scan path (rather than the post-processing of multiple files already acquired in the experimental platform) this is not very useful, as ideally we want to visit each point in the scan region only once, with the laser at the optimal angle for that position. This dichotomy led to the development of the second algorithm, as described in Section 6.2.3.

If some means of ‘pre-scan’ object profiling is used to provide some *a priori* knowledge of relative heights it may be possible to identify scan regions corresponding to specific edges and determine the height difference between the surfaces on either side of the edge. The scan region widths could then be controlled to some extent, extending them more on the ‘low’ side than on the ‘high’ side, although this should not be an issue except where scan regions overlap.

#### **6.2.2.3 Number of Scan Orientations Required for Accurate Representation**

In some cases the orientations of edges within an object are restricted to a limited subset of all possible orientations. For example, the edges in a cuboid object are (from the camera perspective) restricted into two disjoint sets at orientations perpendicular to each other. In

this case only require 2 scan orientations are required rather than the usual full set (of 10 scan files) to capture all the relevant information.

The advantage of this is that the smoothing / edge detection process can cause corners to become rounded, which is then exacerbated by the vectorisation into a straight line that cuts across the corner at an angle between the two major orientations. This ‘false’ vector would then require a partial scan, taking time and resulting in data that does not accurately represent the geometry of the corner (in fact, due to the simple averaging it may make the resulting output values worse). By forcing the system to subsume the orientation of this vector artefact into one or other of the major orientations, the region developed from this vector is merged with the overlapping region from the nearest orientation, which in this case would save time and provide better quality data

This issue can be used to limit the number of orientations considered for a particular object, but it is something that can only be done where the edges in the object fall into sets of easily discernable groups. In objects where curved edges are present, limiting the number of scan orientations is likely to cause errors where the regions change from one orientation to another.

Where there are ‘continuous’ features of high curvature in the object it may be better to select a higher number of scan orientations. However as there is a degree of tolerance in the orientation before errors in the data become significant it may not be necessary to scan at the maximum number of orientations either.

The orientation tolerance (as described in Section 4.1) is somewhat dependent on the particular object under scrutiny and the application for which the scan is being performed, but generally an orientation of anything under  $\pm 9$  degrees (i.e. an 18 degree range) has proven to give acceptable results in testing.

In most cases it does not matter whether the laser emitter or detector is ‘closer’ to the edge. As long as the laser is kept more or less parallel to the geometric edges in scan space, The exception is where the edge is in proximity to internal corners, where the presence of other geometric edges interferes with the sensor reading. For further detail on the Internal Corner issue, see Chapter 8.3.

### 6.2.3 Scan Orientation By Nearest Vector Orientation

The theoretical basis for this algorithm is discussed in Section 4.1.3.2. The practical implementation of this idea is considered here. This method seeks to avoid the issues caused by overlapping scan regions by selecting only the data value associated with the orientation provided by the vector to which it has the closest proximity. This can be done by calculating the distance from the currently considered point to the nearest point on each vector segment.

Whilst this is in principle a simpler proposition than the previous method as it obviates the requirement for determining the average value for a point, it is potentially computationally expensive, especially where large numbers of vectors are present as each scan point must be transformed compared to all vectors. However, it is possible to reduce the number of vectors to which each point must be compared because each geometric edge (as represented by the vector) has a maximum ‘area of influence’, equivalent to the scan region width beyond which it is redundant to consider the edge. The calculation will still be computationally intensive especially in areas where many vectors are in close proximity.

To represent this concept, a ‘coarse scale’ 2D vector map is created with each cell in the map representing a square region of the image of a defined size relative to the scale of the image. Within each map cell a reference to each of the vectors passing through the corresponding region of the image is stored. When the scan point is transformed to the image space, the algorithm discovers within which map cell the corresponding image point lies and only compares the position to the vectors referenced in that cell.

In practice it was found that it was more efficient to further reduce the map cell size and compare each point to a range of cells within a radius that might influence the scan output at that location. Although a vector may pass through a number of map cells, it is only necessary to calculate the distance from the point under consideration to the vector once. If the same vector is encountered in more than one map cell it is not added to the list again, however for tracking purposes (and potentially for the resolution of ‘tied distances’) a count is kept of the number of times each vector is referenced.

Calculating the distance from the point to each vector is simple in the case where the mathematical line from the point perpendicular to the vector falls within the length of the

vector segment, however in many cases this line may fall outside the vector segment, in which case it is necessary to calculate the distance from the point under consideration to the nearest endpoint of the vector segment. In these cases the angle of the mathematical line relative to the vector endpoint is also stored.

### **Algorithm 3 – Partial Scans by Nearest Vector Orientation**

```

set up coarse vector 'map'

determine 'default' scan orientation by sum of vector lengths by
orientation

write base output scan according to default scan orientation

for each point in output scan image
    transform to image coordinates
    determine corresponding vector map cell

        for each map cell within defined radius of current cell
            for each vector referenced by that cell
                if vector index already added to vector list
                    increment count of occurrence
                else
                    add vector index to list
                    calculate distance and angle
                    from current point to nearest point on vector

sort vector list by distance
(in the event of a tie, use angle to decide which is closer)

get closest vector orientation
look up matching orientation data file at current point
write selected data point to output file

```

Once all the relevant vectors have been processed they are sorted by distance from the current point. In the event of a tie, the angle between the vector and the point is considered; if the mathematical line from the point perpendicular to the segment falls within the length of vector A and outside the length of vector B then vector A takes precedence. If the current point falls within the length of two or more vectors and is equidistant between them and no other vector is closer then the vector encountered more often within the radius of the current pixel (i.e. the vector reference count) is used to decide which orientation is more important. The orientation of the vector determined to be closest to the current point is taken to be the scan orientation for that point.

In general it was found that this method produced better results than the previous Scan Region method, however as stated in Chapter 4, there are potential issues with this method. Each point has only one original value representing the height and if the laser sensor is not well-calibrated there may be differences both in the relative position of edges between scans at different orientations and the perceived height of these edges, resulting in sudden discontinuities in the data. In the test program it was decided not to implement any smoothing or region blending to disguise the potential extent of these problems. However for use in a commercial system some means of blending between partial scan regions would be advised.

Further development of both the above scan region methods was considered but not implemented. These related mostly to scanning strategies that may be applied in generating a scan-path control output file for a working system rather than the test system which requires a number of pre-defined scan orientation files as input.

These considerations included a method by which regions could be expanded to fill non-critical areas of the scan region and to join adjacent (but non contiguous) regions of the same orientation by a technique similar to the 'region growing' method applied in image processing to segment images. This would speed up the overall scan time by allowing scan lines to continue instead of starting and stopping short runs which would require constant acceleration / deceleration of the laser sensor. Conflicting data where 'joined' scan regions overlapped with an actual recognised region could then be discarded as a post scan processing stage. However, as the test system required a number of pre-defined scan files as input rather than producing a single scan control path file as output it was not really possible to develop this idea further.

### **6.3 Scan Data Interpretation**

In the experimental system it was necessary to use pre-scanned data files as the Scan3D software used to control the movement of the laser sensor cannot support the processes required. It is not able to control the rotation of the sensor, integrate several partial scans to provide a composite output point cloud, nor is it able to accept an externally produced scan path file as control input at this time.

The use of pre-scanned point cloud data files does have the advantage (from the perspective of this project) of allowing the production of ‘virtual scan’ output files. That is a resultant point cloud output that approximates the intended result of controlling the scan path can be produced without having to perform the scans to test each scenario.

As the scanning operation is by far the most time-consuming aspect of the whole process, the ability to re-use a set of scans for a given object is advantageous in testing the effects of changing either the software algorithms applied or simply the parameters used to determine the output of each stage in the process without requiring that the object is physically rescanned each time.

However the change in emphasis of the test program from producing a scan path control file to producing a combined point cloud representation requires that the test program be able to accept a set of files representing the object under consideration at a number of different orientations and be able to reference common points within that set with reference to the output of the image analysis.

The scan data output for each scan orientation produced by the Scan3D program was saved using the standard gCode file type and a common naming structure. The set of files for a single object is then loaded into the image processing test software as described below.

### **6.3.1 Loading - Data Regularisation / Interpolation**

Each scan data file consists of header data, showing the extents of the scan area (in the CNC machine coordinate system). Some modifications to the standard file header data are required in order to indicate the scan resolution and sensor orientation. The header information also includes file path information, version number and modification date. All the header information is parsed and stored within the test program.

The resolution of the scan in millimetres is required in conjunction with the extents order to determine the number of scan lines and data points per scan line. It is assumed that the resolution is the same in both x and y dimensions in the scan data.

It was decided that a scan orientation of 0 degrees would indicate the laser sensor being parallel to the y-axis of the CNC machine, with positive increments indicating a counter-clockwise rotation.

The standard gCode file does not present the data on a regularised grid. Data is stored relative to an initial point and although each row is a consistent separation, the data points along each row are not necessarily a consistent distance apart due to the way the laser sampling works. The gCode data only stores the values that have changed since the previous data sample, so the Y value only appears at the start of each new row. Also if the Z value (i.e. the height information) is constant between adjacent sample points the Z data value does not appear in the file.

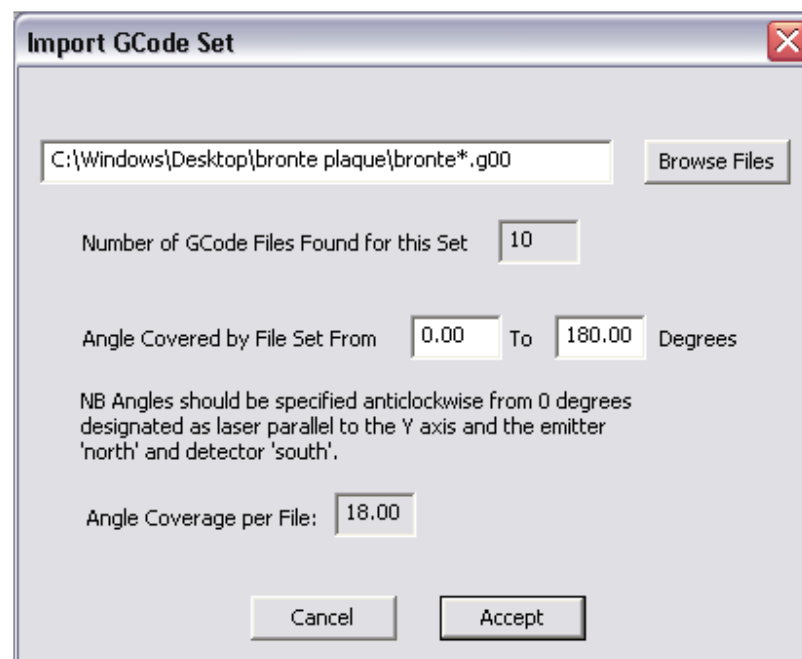
Within the image analysis test program the same data structure that is used to represent the edge image data is also used to store the scan data (i.e. a 2D array of floating point values). In order to read the data correctly it is necessary to interpolate the data points into a two-dimensional array. The data is read from the gCode file until a whole row has been accepted, then the data for that row is fitted to the array using a linear interpolation.



### 6.3.2 Display / Operator Interaction

The operator is required to select the correct set of gCode files for the current object. This is done via a standard Windows file handling dialog box. The functionality allows for the 'wildcard' ('\*' and '?') characters to be used to match multiple files, and the number of files matching the current file path is displayed. All the files matching the current path will be imported as a set at one time.

A standardised file naming convention has been used whereby the initial part of the file name is consistent for all the files in a set, with the specific orientation angle for each file tagged onto the end of the file name. It is assumed that all the files within the set are equally distributed between the minimum and maximum orientation angles selected by the operator.



**Figure 6-6: GCode File Set Import Dialog allows selection of multiple files representing a range of sensor orientations**

### **6.3.3 Scan Time Estimation**

The scan time is one of the criteria used to determine the level of success in this work. Wong's method provided good results, but required a set of complete scans representing the whole object at 8 orientations of 45 degree separation.

Even a relatively small (e.g. 5cm square) object scanned at a resolution of 0.01mm can take about 2 hours to scan for each orientation with the test rig, which may render the method commercially uneconomical. The partial scan method brings down the total required time because only a part of the image is scanned at each required orientation, however, due to the fact that scan regions often overlap, the total scan time will almost certainly be greater than that for a single pass scan.

The total scan time for the complete image can be estimated, based on the scan resolution and the speed of the scanning head movement over the object. The time for each partial scan can be calculated according to the scan area covered within each orientation, and the sum of the partial scan times plus an overhead to allow for the time taken to rotate the laser between orientations and re-calibrate the position of the laser spot (if necessary).

Efficiency improvement could be made by 'subtraction' of scan orientation areas from default scan direction. However, this could not be done with the test platform, because all the scans were performed prior to the determination of scan region coverage, which would not be required in a system that generated output in the form of a scan control path file.

## **6.4 Summary**

This Chapter shows how the ideas described in Chapter 4 may be realised by algorithms for each of the approaches suggested. The framework of the ‘proof of concept’ test software is described and the scan acquisition process is detailed in order that the experiments may be replicated accurately. The process of calibrating the images to the scan data is described, as without accurate calibration it is not possible to relate the scan files to the results of the image analysis process. The practical concerns of interpreting the scan data files in a form that is compatible with the image analysis software are also addressed.

The main algorithms implemented for testing are the ‘Partial Scan by Scan Region’ and ‘Nearest Vector Orientation’ methods. The ‘Dominant Scan Orientation’ method is also implemented and used as a base to which the other methods are compared. The final method described in Chapter 4.1.4 was not implemented due to hardware limitations.

There are still a number of situations that require further investigation, such as methods of deciding the best means of resolving overlapping regions and rounded corners. Corner detection at the image analysis stage may help to provide further information for these tasks; however of the many corner detection algorithms available none are reliable in all situations.

The detection of corner features could be used to apply Wong’s method to local area around the corner. This would improve the scan time as it would be possible to limit the areas that required duplicate orientations. It may also be possible to reduce the set of orientations required by Wong’s compensation method where the orientation of edges is known in advance.

As an extension to the method for low detail regions (as determined by the lack of evidence for any edges or texture), it may be possible to reduce scan line density (thereby allowing faster scanning). This is subject to limitations as there may still be variation in surface height (e.g. a gently curving surface) that is not well-recognised by edge detection.

The results of applying these algorithms to a number of different objects are presented in Chapter 7.

## Chapter 7 – Experimental Results

On completion of the scanning process the output range images produced by the two ‘partial scan orientation’ methods were examined for evidence of scan distortions. The results of the two combined scan methods are compared with the range image determined to be the ‘optimal orientation’ that could be achieved in a single scan.

The choice of objects was limited somewhat by the working range of the available sensor, which is only 2cm. Because it is also difficult to study the causes of distortions when objects have complex details, initially simple objects with primitive geometry were used, e.g. rectangular blocks (sometimes with rounded corners) or circular blocks.

Each object was lit from the left and right with diffused 40W fluorescent tube lights and a white reflector was placed over the object to reflect light onto the object from above. This provided a soft directional lighting setup that was found to minimise the effects of shadows on edge detection. (See Chapter 4, Figure 4-9 for details of the hardware set up).

The PC used for the image processing, determination of scan regions and subsequent production of the combined output files was consistent throughout the development process so that timings between scans would be comparable: the PC was equipped with an AMD Athlon 2800 CPU running at 2GHz, with 1GB of RAM. The operating system on the PC was Windows XP.

Objects with more complex geometry were then used to evaluate the algorithms developed on more ‘realistic’ surfaces. This chapter presents a progression of experimental results for several objects on which the different scan region algorithms developed in Chapter 6 were tested.

The objects used for testing were as follows:

1. Domino – this is a generally rectangular shape with rounded corners and the surface has an embossed detail of a greyhound. The domino was tested first with the surface uncoated, and then with the surface sprayed with Flawfinder Developer to observe the difference in results.
2. Bottle Top – this object was used to determine how well the algorithm copes with arcs of a fairly low curvature. The bottletop was sprayed with Flawfinder Developer to provide a diffuse reflective surface.
3. Gauge Plate – this is a metal block of known dimensions. It provides a simple (almost rectangular) profile, however the size of the block presents a problem with the calibration. The plate was sprayed with Flawfinder Developer to reduce specular reflection.
4. Triangle block – this is a simple triangular metal shape, however it contains a linear engraving and a hole in the surface with a high curvature that was expected to cause problems. The block was sprayed with Flawfinder Developer to reduce specular reflection.
5. Cookie Cutter Mould – the giraffe cookie cutter mould contains many edges at different angles and provides a challenging ‘real world’ object to scan. Again, the mould was sprayed with Flawfinder Developer.
6. Brontë Plaque – this is a very complex ‘real world’ object with much detail at different scales. The plaque was *not* sprayed because the surface was already a matte finish and the spray is hard to remove from small texture details in the surface.

The relative times for processing and scanning each of the objects are shown in Figure 7-1. Scan times increase in proportion to the resolution (rather than the square of the resolution) because the sample speed of the laser is much higher than the speed of the scanner gantry movement. The translation in the x-axis is therefore controlled by the speed of the machine movement, so the scan time is affected by the length of each scan line rather than the scan resolution in the x-axis.

Object	Surface Treated	Scan Area (mm)	Time to process 640x480 8-bit image (seconds)	Time to Scan (min:sec) per orientation @ resolution:		
				0.5mm	0.1mm	0.05mm
Domino (1)	No	60 x 50	ED 1.06 V 0.35 SRO 2.0 NVO 7.2	7:50	39:10	78:20
Domino (2)	Yes	60 x 50	ED 1.13 V 6.50 SRO 2.03 NVO 29.3			
Gauge Plate (webcam)	Yes	120 x 80	ED 1.29 V 0.5 SRO 0.5 NVO 9.45	19:12	96:00	192:00
Gauge Plate (digicam)	Yes	120 x 80	ED 1.2 V 5.48 SRO 0.57 NVO 9.29			
Bottle Top	Yes	60 x 60	ED 0.91 V 0.94 SRO 2.05 NVO 10.15	9:24	47:00	94:00
Triangle Block	Yes	60 x 60	ED 0.832 V 1.6 SRO 1.99 NVO 10.26			
Cookie Cutter	Yes	110 x 70	ED 1.08 V 63.74 SRO 1.75 NVO 244.47	15:40	80:30	158:40
Bronte Plaque	No	120 x 90	ED 0.95 V 138.67 SRO 2.90 NVO 433.51	21:36	108:00	216:00

**Figure 7-1: Comparison of Processing and Scan Times for the Objects Used in Testing.**

ED = edge detect using Canny detector

V = vectorise using Least Squares method

SRO = create scan region map (10 orientation groups)

NVO = create nearest vector map (10 orientation groups)

The y-axis is the scan line separation which is affected by the resolution (higher resolutions obviously require more scan lines with smaller separation). In addition there is a constant overhead for each scan line i.e. acceleration/deceleration phase and machine slew (set to start position for next line). For small objects (such as these test objects) where the length of each scan line is relatively short, the overhead takes up a greater proportion of the total scan time.

### ***7.1 Methods for Evaluation of Results***

Determining the quality of the results of the scan region algorithms is a somewhat subjective task. It is difficult to quantify the accuracy of the process. Obviously a situation in which no errors occur is the ideal, but the problem is how we can establish the 'benchmark' to which we compare the scan results. Given the nature of the scanner, in almost all 'real object' situations there will always be some errors at any scan orientation. It is also debateable whether a few large errors are better or worse than many small errors. A large error may be easily identified as such (and therefore post-processing compensation can be applied more easily to eliminate or reduce the error), whereas a number of smaller errors may not be as separable from the correct scan data. Therefore assessing the accuracy of the composite scan output is primarily a qualitative assessment, as the total number or magnitude of errors does not necessarily equate to a visibly improved scan. As with the assessment of edge detection algorithms, a visual assessment by human operators is possibly the best measure.

Graphical output is provided for each object with the z-axis magnified with respect to the x and y axis in order that errors can be more clearly observed. The z-axis in each case is scaled in millimetres. The x and y axes are represented by one sample point per 0.5mm. This coarse resolution is sufficient to demonstrate the occurrence (and removal) of occlusion and reflection errors without requiring excessively long scan times. One method that can be applied with some degree of success is to plot the difference between the scan determined to be the 'best' single scan orientation and the composite region scan. This will highlight where the composite output is better (or worse) than the best single scan.

The discussion of the results is divided into a number of sections that discuss the process and highlight particular issues that were discovered during development of the methods. Section 7.2 provides an overview of the practical scan acquisition process, from image capture to the production of the scan region maps. Section 7.3 discusses the effects of surface reflectivity on the quality of the scan data. Section 7.4 highlights issues relating to optical distortions caused by the camera. Section 7.5 discusses the issues of overlapping regions using the Scan Region Orientation method. Section 7.6 demonstrates the ability of the methods to cope with regions of curvature, from smooth, low curvature to sharp corners and Section 7.7 shows the application of the method to ‘real-world’ objects.

## ***7.2 Example of the Scan Acquisition Process***

The process of scanning an object follows the process outlined in Section 6.1.1. An example of the process of creating a scan region map for a given object is described here. The back face of a domino (shown in Figure 7-2) is used as an initial test object.

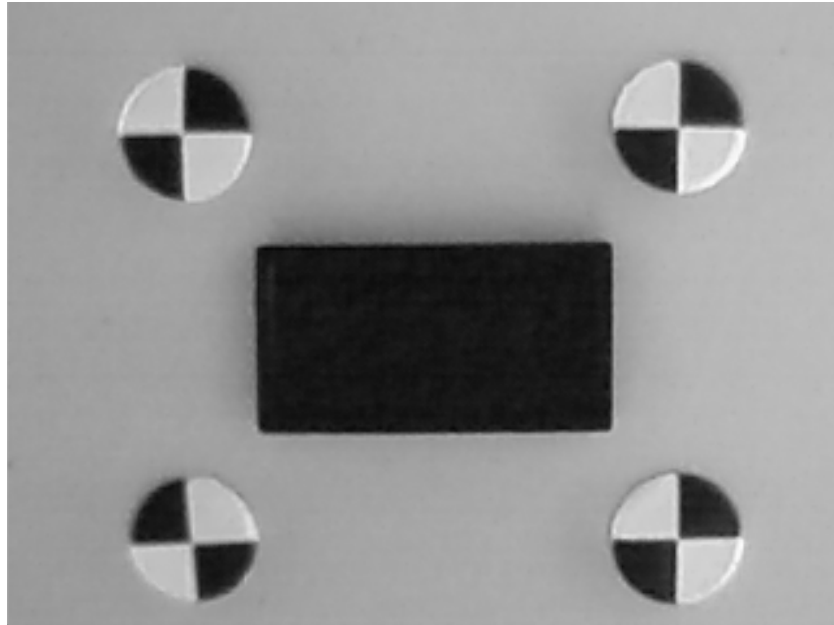


**Figure 7-2: Picture of domino object showing embossed pattern on back surface. This picture was taken with a higher resolution digital camera rather than the webcam used for the edge detection**



### 7.2.1 Domino Calibration

The scan region area is a 60mm x 50mm area defined by the centres of the calibration markers around the domino in Figure 7-3. The scan resolution used in this test case is 0.5mm. In the image space the resolution is approximately 0.16mm per pixel.

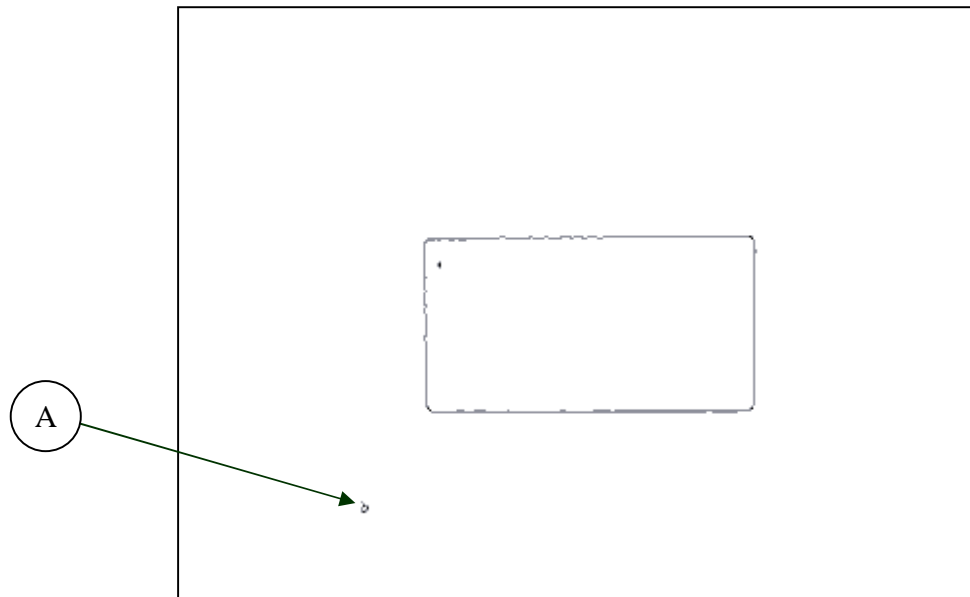


**Figure 7-3: Domino showing calibration markers placed around the object.**

The pixel coordinates of the centre points of each calibration marker and their corresponding locations in scan space are entered into the software to provide scaling, skew and rotation factors for the calculation of the scan regions. This allows any position in the image space to be related to its corresponding position in the scan coordinate system, thus the ‘correct’ data associated with any given orientation scan can be selected to create the composite output scan. Once the image is calibrated, the markers are removed and the image captured again without disturbing the object or camera position, so that the markers do not register as edges at the image processing stage.

### 7.2.2 Domino Edge Detection Stage

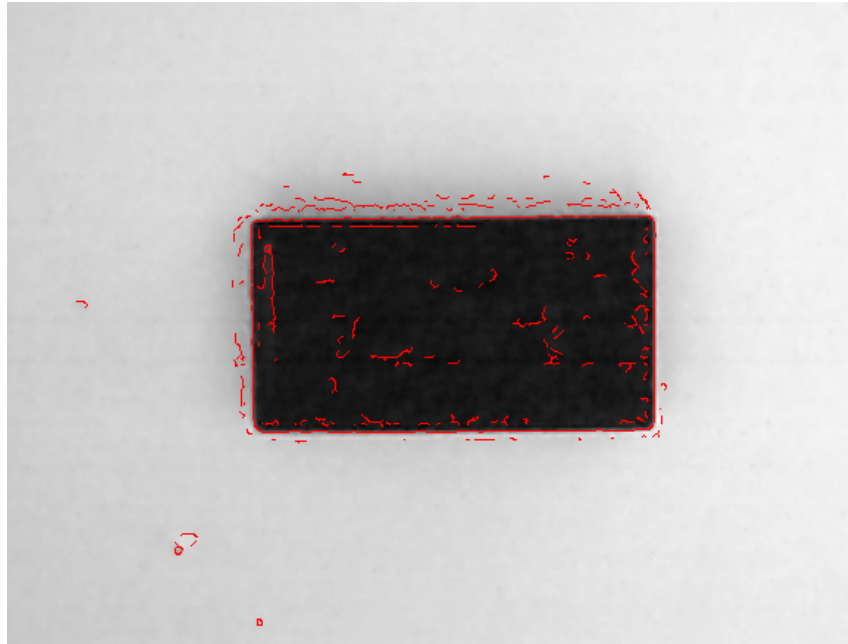
The Canny edge detector is used to determine the edge detection for this object with Gaussian smoothing  $\sigma$  1.0 and thresholds of 97% (max) and 27% (min). The time to process this 640x480 pixel image was 1.06 seconds. The results of the edge detection can be seen in Figure 7-4.



**Figure 7-4: Edge Detection stage showing results from Canny edge detector. The small circular spot (marked by arrow A) detected in the bottom left of the image is an alignment mark for the laser.**

Edge detection shows good separation of the object geometry from the background with the selected parameters, however the details on the top surface have been lost. It is not possible to isolate the detail of the embossed shape from the background noise level in the image. However, as the embossed area contains no sharp edges and the height variation over the surface is less than 1mm this was not expected to cause a problem with data distortions.

Relaxing the selection parameters by selecting a lower smoothing value or lower maximum threshold level allows many 'false positive' edges to be detected as shown in Figure 7-5. Finding the best balance of smoothing and thresholding levels is a matter of trial and error for any given image, although settings remain approximately the same as long as the lighting conditions are not changed too much between different objects.



**Figure 7-5: Edge Detection overlaid on uncoated domino with slightly lower threshold settings results in detection of numerous false positive edges**

### **7.2.3 Domino - Vectorisation Stage**

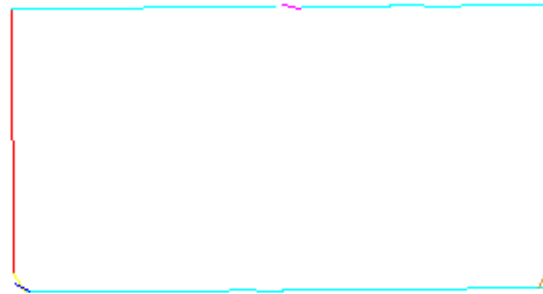
Performing least-squares vectorisation (see Chapter 5.2.2.2) on the results of the previous edge-detection stage reduces the image to a small number of line segments that closely match the outline of the original image. The initial parameters used to produce the output shown in Figure 7-6 were as follows:

Min Allowed Segment Length 7 pixels

Line Angle Delta Tolerance 9 degrees

Bridge Gap Limit 5 pixels

Execution of this stage with these parameters was performed in 502ms.



**Figure 7-6: Vectorisation of Domino (edge only)**  
**Note small discontinuity in top edge and short vectors at bottom corners**

Short ‘noise’ edges are removed automatically by the algorithm as they do not meet the minimum vector length criteria. A slight discontinuity can be seen in the top edge of the image, however this can be attributed to the vectorisation step following a perturbation in the detected edge. Note also that short vectors have been discovered at the bottom corners of the domino where the appearance of the corner in the edge-detected image is slightly more rounded than on the top corners. This is because the camera was not centred over the domino, thus giving a slightly off-centre view of the object.

These artefacts can be removed by selecting a different set of parameters for vector length and angle tolerance. Selecting for a longer minimum allowed edge length and a more relaxed angle tolerance causes the short edges to be subsumed into longer runs, causing the discontinuity along the top edge to be merged into the longer vector.

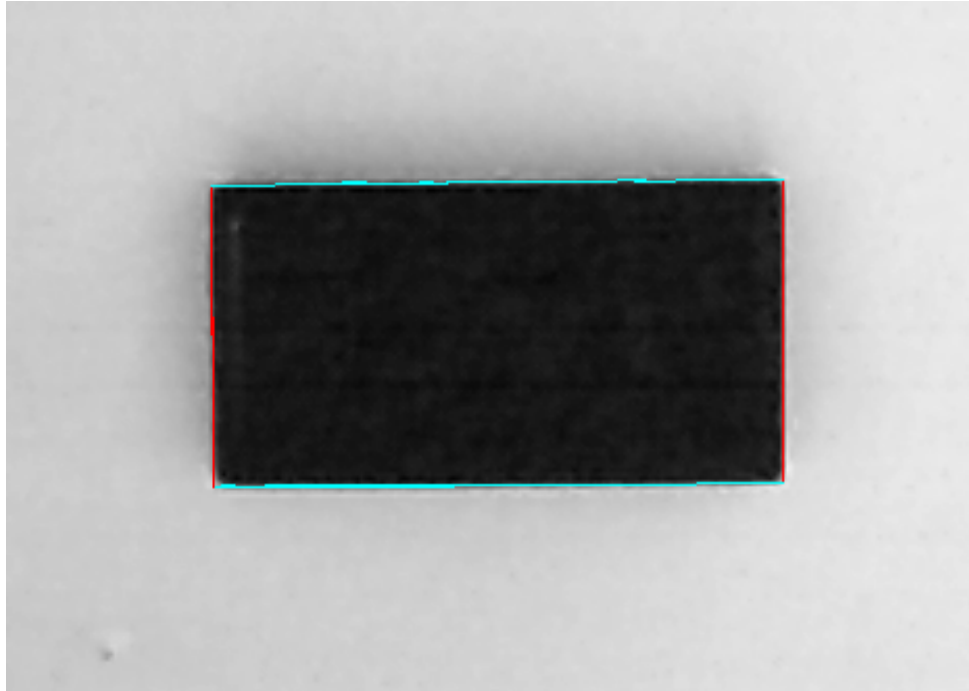
The following parameters were used to produce the output in Figure 7-7:

Min Allowed Segment Length 11 pixels

Line Angle Delta Tolerance 24 degrees

Bridge Gap Size < 10 pixels

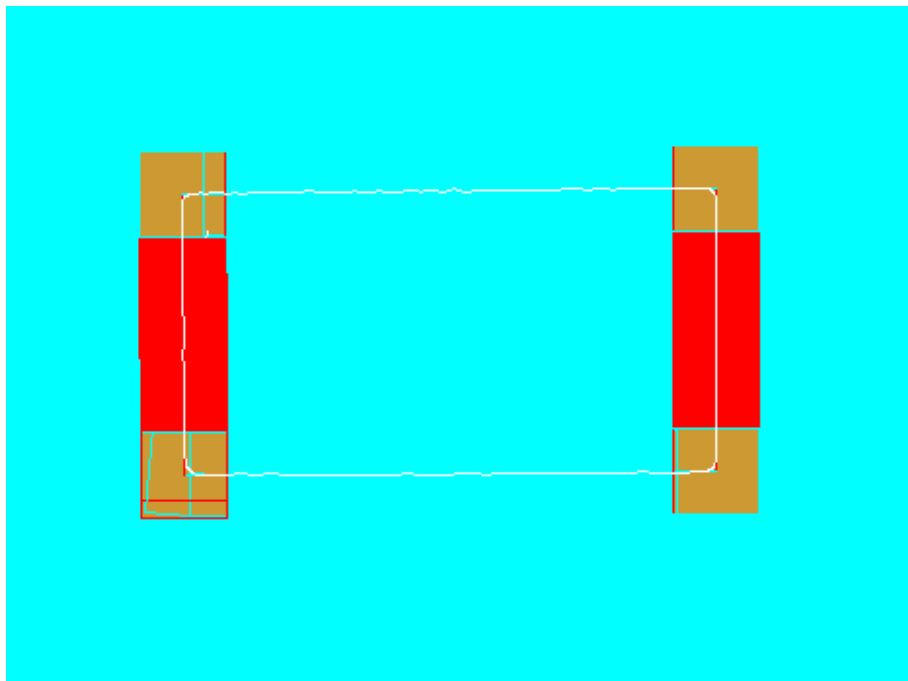
With these parameters the short vectors ‘allowed’ in the previous set have been discarded and the more relaxed angle tolerance has allowed the merging of the short edge vectors with longer vectors, providing a better overall match to the outline of the domino . The results shown in Figure 7-7 are overlaid on the original picture for comparison of edge localisation. Execution of the vectorisation stage with these parameters was 350ms.



**Figure 7-7: Domino with Vectorisation Overlay. Changing the parameters for the vectorisation step has removed the discontinuity in the top edge and the short corner vectors.**

### 7.2.4 Domino – Scan Selection by Region Orientation

The scan region combination method produces the output map shown in Figure 7-8 based on the vector map shown in Figure 7-7. This shows that the combined scan will select from the dominant ( $90^\circ$ ) scan direction as indicated by the cyan area for the majority of the image. Along the shorter edges in the image it will select from the  $0^\circ$  scan as indicated by the red areas. In the corners, where the regions overlap, (as indicated by the brown areas) the values from both areas are selected and a simple average of the values is taken. In selecting these areas there is no ‘blend’ or smoothing between neighbouring regions in this algorithm. Although this may improve the appearance of the overall output, it may also disguise the problems encountered in using this method. The problems with overlapping regions are discussed in Section 7.5.

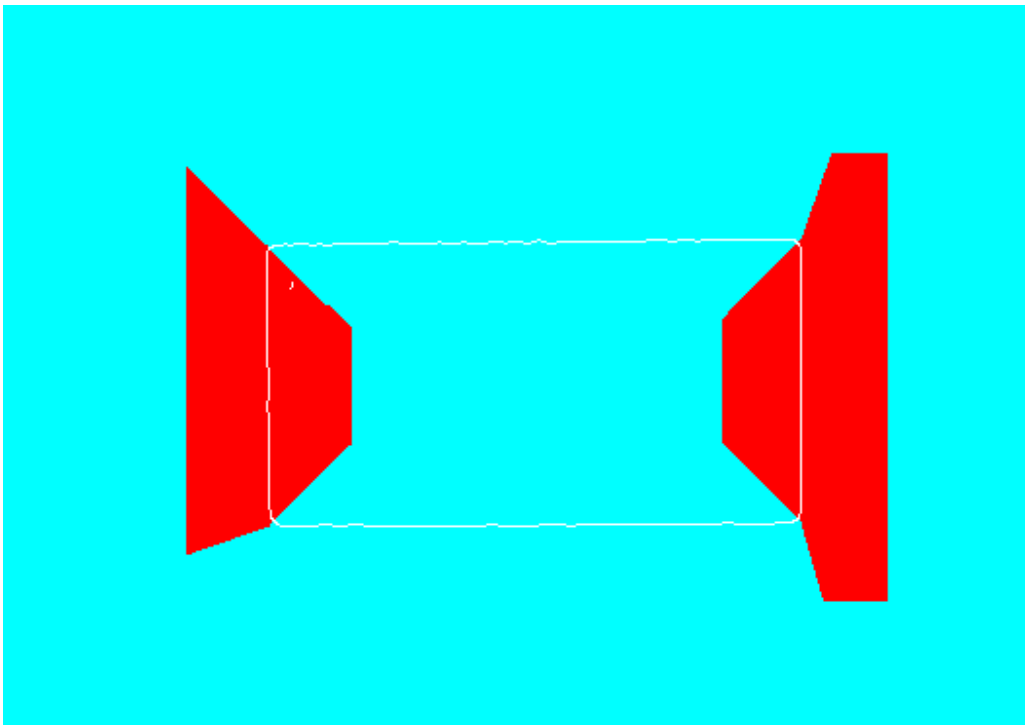


**Figure 7-8: Output map for Method 2 (Combine Scans By Region Orientation) on domino (edge detection overlay to indicate position of domino)**

### 7.2.5 Domino – Scan Selection by Nearest Vector Orientation

In this method, for each point within the overall scan area, the selected orientation data is that which most closely matches the orientation of the nearest vector. This gives a single value for each point and no averaging of values is required.

This algorithm is also conceptually closer in terms of what is required to the original problem of developing a laser scanning path (e.g. each point in the scan is visited only once). The output scan region map by this method is shown in Figure 7-9, based on the vectorisation map shown in Figure 7-7. Execution of this stage took approximately 7.2 seconds.



**Figure 7-9 : Scan region output map for Method 3 (nearest vector). The original Canny edge detection result is overlaid in white for reference**

Ideally, the two 0° scan areas (indicated in red) should be ‘mirror images’, however due to the slight differences in the overlap of vectors at the corners and the effect of overlaying the ‘vector grid’ (see Section 6.2.3) on the image causes the regions to appear as shown.

### **7.3 Surface Reflectivity Issues**

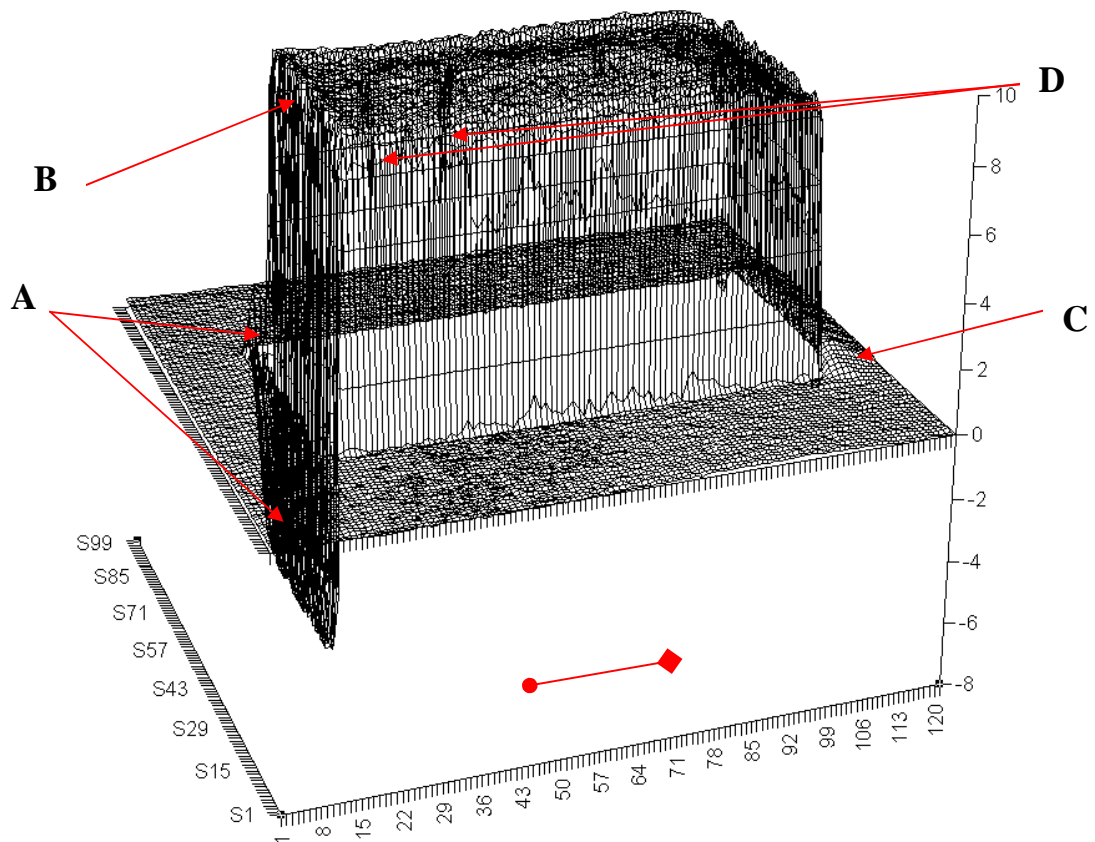
The surface reflectivity of the object being scanned makes a significant difference to the quality of the scan output. Initially, the domino (as described in Section 7.2) was left unsprayed in order to test the quality of the scan from an uncoated, matte surface of low reflectivity.

The single 'best' direction in which to scan this image as determined by the sum of the vector lengths per orientation is with the scanner parallel to the x-axis, which agrees with human assessment of the object in order to keep the scanner parallel to the longer edges. However, as expected, this causes major trough / spike and bow wave anomalies to occur along the edge perpendicular to the scan orientation (see Figure 7-10). Significant spikes and troughs (marked by arrow A) can be observed in the scan data caused by occlusion of the scanner. A spike effect along the top edge of the domino (marked by arrow B) is also present. On the opposite side of the graph a bow wave effect can be seen (marked by arrow C). The scan is relatively good along the longer sides of the domino and within the area of the top although the detail of the pattern is difficult to see. This is in part due to the relatively low scan resolution used in testing and also the very low reflectivity of the matte black surface of the domino.

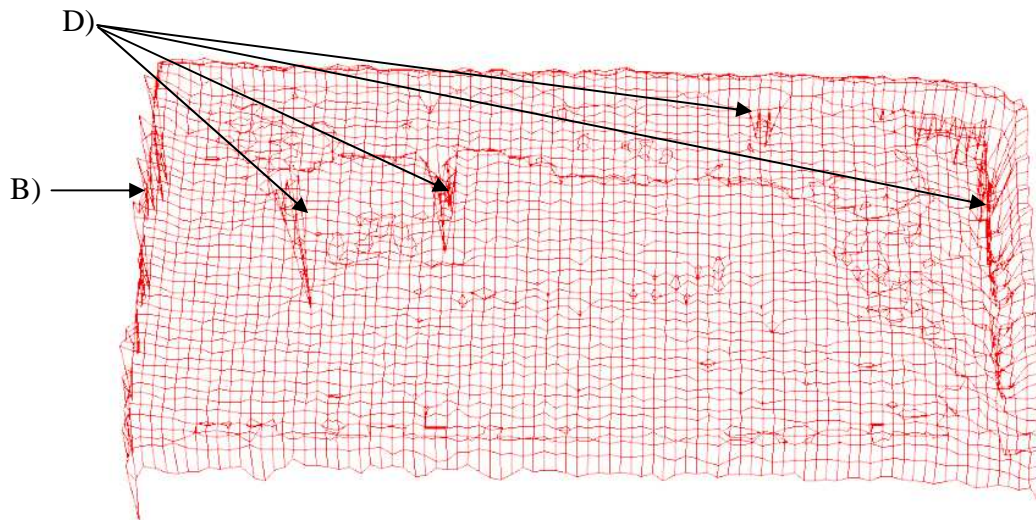
More difficult to distinguish in Figure 7-10 are the 'holes' in the face of the domino (marked by arrows D). These are shown in greater detail by the excerpt from the scan area shown in Figure 7-11a). These may be caused by small localized occlusions of the surface by the shape of the embossed greyhound around the 'collar' and 'nose' area. Given the limited depth of the embossing it was surprising to see these spikes in the data, but this shows the sensitivity of the system to occlusions, especially on surfaces where the level of reflection (from the matte black finish) is very low. Any combined partial scan method will carry forward these errors caused by the low reflectivity. When the surface is sprayed with the Flawfinder Developer spray, the increased reflectivity of the surface provides a much better response from the laser (as shown in Figure 7-11b).

The improvement shown in the scan quality by spraying the object in cases where the surface reflectivity is low is clear (see Figure 7-10 and 7-11). When the domino was re-tested with a sprayed surface, however a problem with overlapping scan regions was found, which is described in Section 7.4.

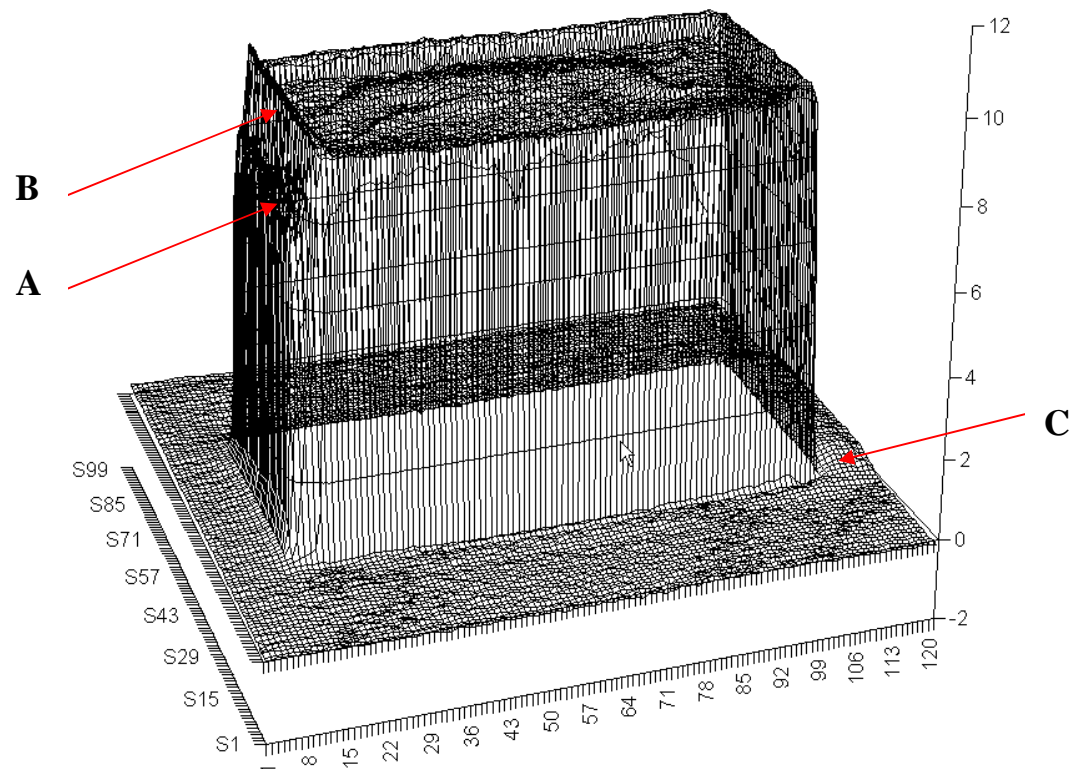




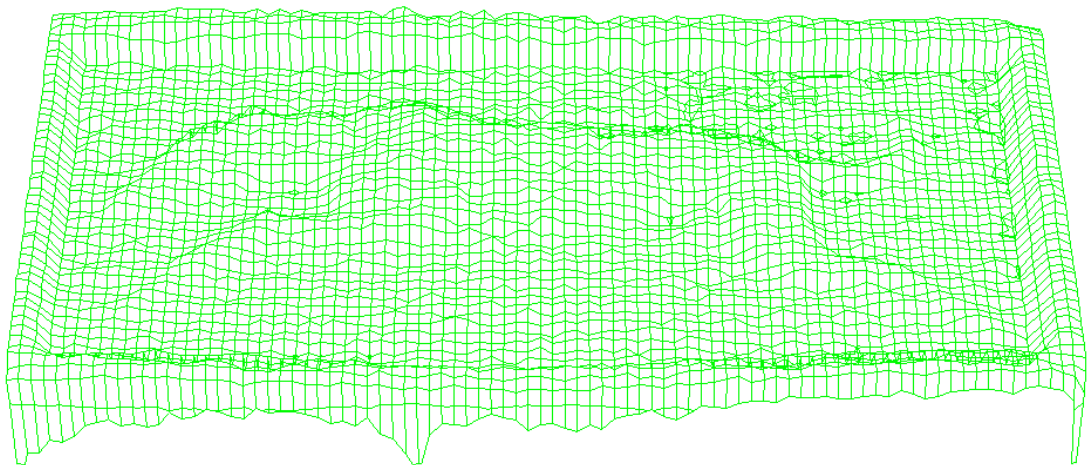
**Figure 7-10a): Dominant Direction Scan of Domino with Surface Unsprayed. The laser orientation is parallel to the x-axis, as indicated by the red line (with the diamond end as the detector and the circular end as the emitter). Observable errors are marked A-D are discussed in the text**



**Figure 7-10b): Excerpt of Domino Surface Mesh from dominant direction scan. Arrows labeled B & D indicate Occlusion Spikes as above.**



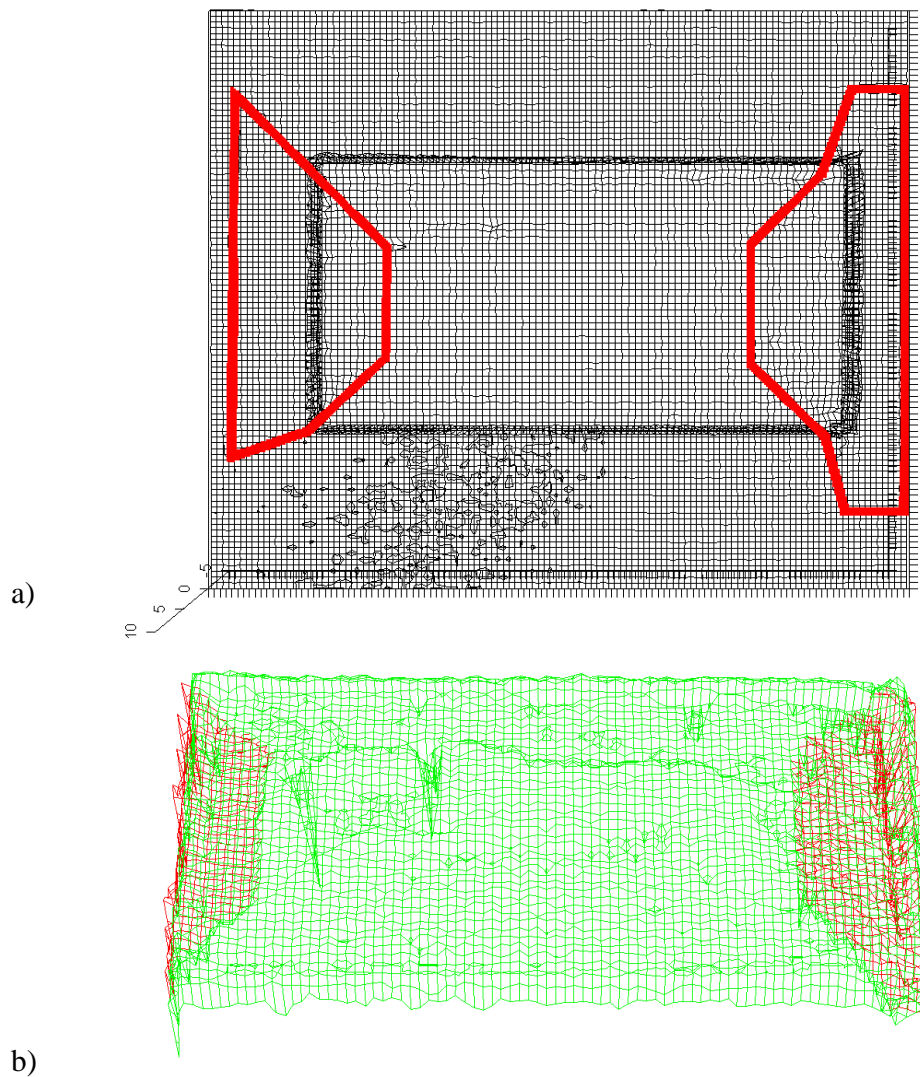
**Figure 7-11a): Dominant Direction Scan of Domino with Surface Sprayed.** (laser orientation as above). Observable errors A& B are much reduced and D is completely eliminated, however the bow wave (C) is increased due to the increased level of reflectivity.



**Figure 7-11b): Excerpt from surface mesh from dominant direction scan of domino coated with Flawfinder Developer spray.** Occlusion Spikes within the surface are no longer found and greyhound profile is better defined.

## 7.4 Distortions Due to Camera Alignment and Optics

Unless the object is very small, image distortions caused by the camera optics are likely to cause misalignment of the scan regions. For the domino scans it became apparent that the image scan areas are slightly misaligned with the edges of the domino in the scan data. This can be seen in Figure 7-12 where the scan regions are shifted to the left relative to the actual object position.



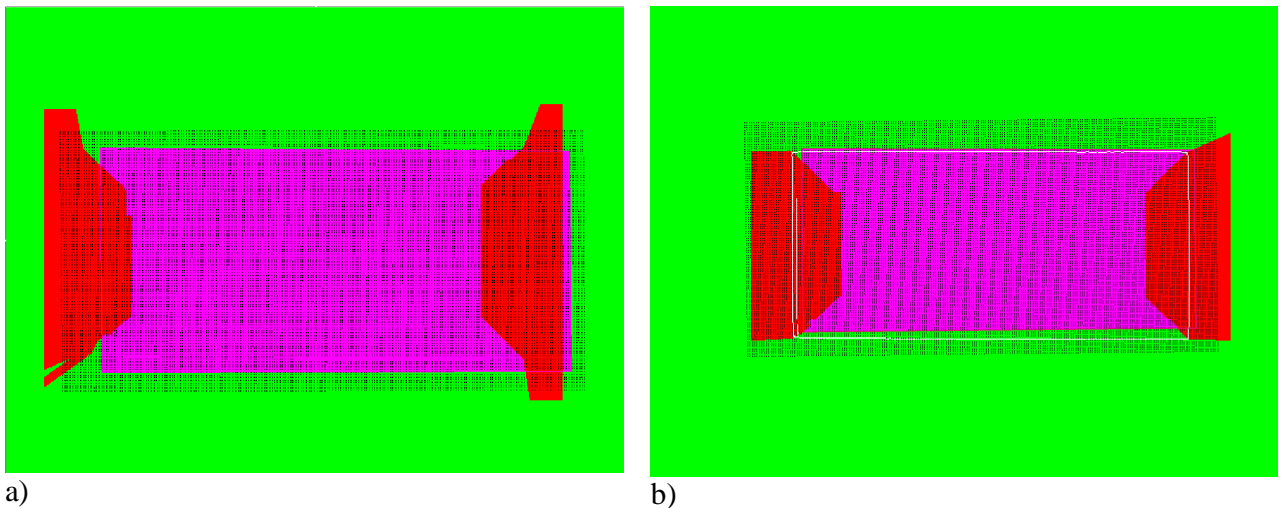
**Figure 7-12:** (a) Nearest Vector Orientation on domino shows poor alignment of regions (red outline). Regions generated by the scan map (shown in Fig 7-9) are not correctly aligned with the object's position in scan space. (b) Detail of combined scan top surface of domino (in green) overlaid on the default 90° scan (in red), shows that the positions of the regions selected from the 0° scan are shifted left relative to the scan coordinates.

The scan map is generated independently of the laser scans (based on the camera image). The problem is apparent in relating the scan data with the data generated from the image. This ‘delocation’ problem is barely noticeable in regions that are relatively central in the image space, but becomes more apparent in regions that are located in the periphery of the image. These issues were further investigated using the ‘gauge plate’ object as described in the following section.

#### 7.4.1 Gauge Plate

When regions are selected from the corresponding scan data files to compose the output based on the image information, it was found that the image scan regions did not correspond well to the positions of those edges in scan-space. The degree of the observed error in the combined scan output region was more noticeable than for the domino because the gauge plate is larger.

The steel gauge plate provides a simple, almost rectangular, block of known dimensions (80.2 mm by 38.3mm) and precisely 8mm in thickness (i.e. height) and the surfaces are flat and diffusely reflective (once sprayed with the Flawfinder spray). The gauge block was imaged using both the webcam and digital camera for the purpose of comparing the two cameras.



**Figure 7-13: NVO scan regions (red areas) overlaid on ‘true’ position of the gauge plate within the scan space (represented by the magenta rectangle): a) the scan regions determined using the webcam image; b) the scan region positions determined from the digital camera image. Selected scan points (indicated by the black dots) show an offset between the calculated position of the scan regions and the actual position of the object.**

The initial calibration, image capture and edge-vectorisation process was followed as described in Section 7-2 for both cameras, resulting in the scan region maps shown in Figure 7-13. The black dots represent the coordinates sampled in the image space corresponding to each selected point in scan space and the magenta area represents the position of the block within the 90 degree (default) scan space.

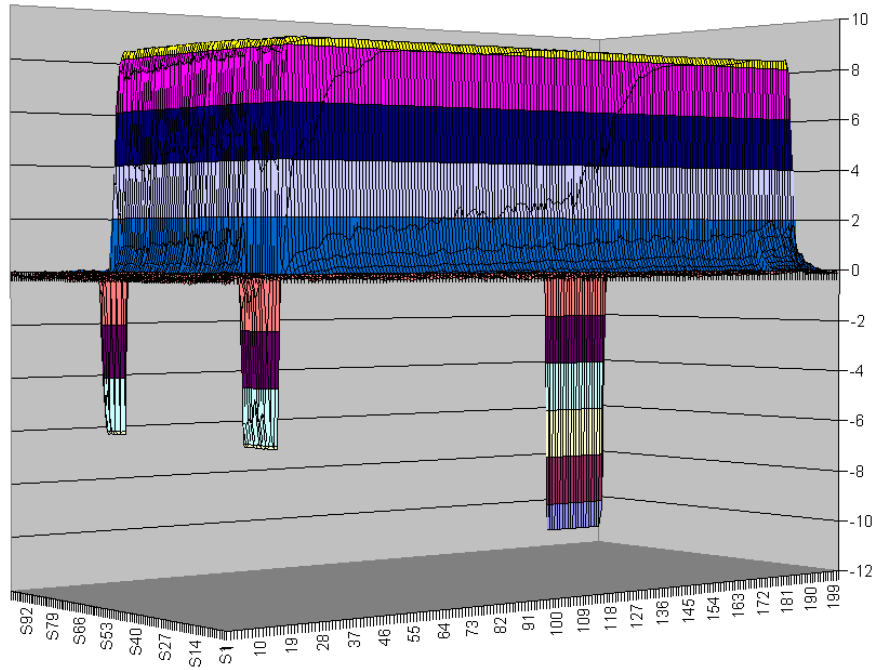
Figure 7-13a) shows the result of applying the nearest vector method using the webcam and Figure 7-13b) shows the result of the same method using the digital camera (variations in the size and shape of the scan regions are due to the different initial conditions for the edge detection and vectorisation).

In both images an offset between the calculated positions of the partial scan regions (relative to the actual position of the gauge block in scan space) can clearly be seen. However the misalignment of the webcam image (left) is considerably worse than that of the digital camera. The left edge scan region is shifted left and only partially intersects the edge with which it is supposed to be aligned. The offset of the image region along the right edge from the correct position in scan space is more pronounced. The region occurs too far to the left and it ‘misses’ the edge completely. These errors render the combined scan output unacceptable (as shown in Figure 7-14).

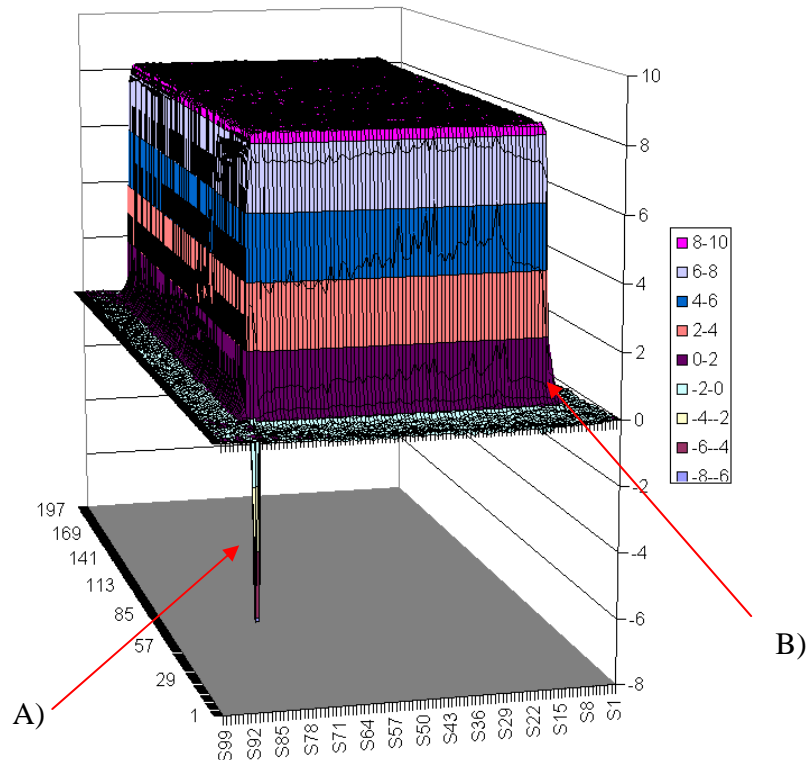
In comparison, misalignment of the scan regions using the digital camera image is significantly less than the webcam (although still imperfect). Location of the corners is much closer than in the webcam image, with just a single, localised trough spike due to misalignment of the grid. Figure 7-15 shows the output from this method.

The corner spike A) (corresponding to the top left image corner) is due to selecting points from the 90° image where it should be selecting from the 0° image. A less visible error is the ‘bow wave’ (B) which causes the edge to appear ‘buttressed’ where the corner alignment misses the correct position by several sample points.





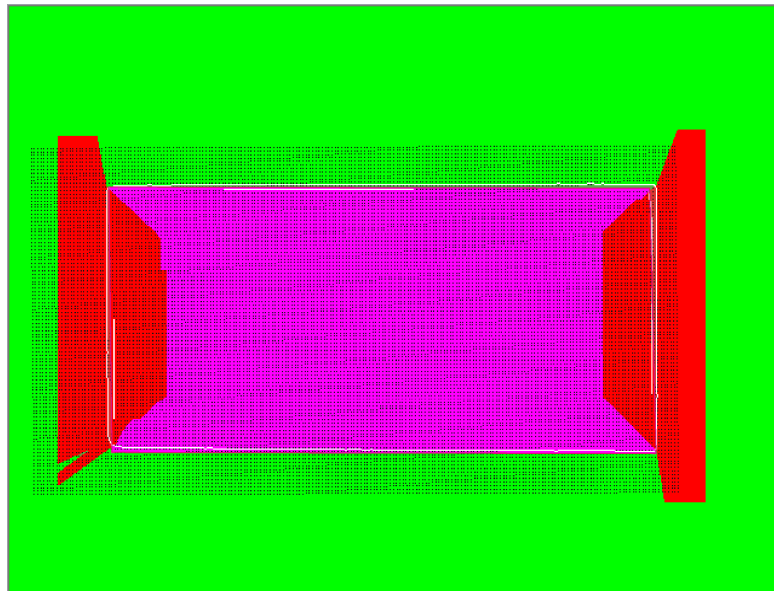
**Figure 7-14: Output from Nearest Vector Orientation Method using webcam for image capture. Note that the magnitudes of 'trough' errors are different in the two original sets of scan data, which is why they appear different here.**



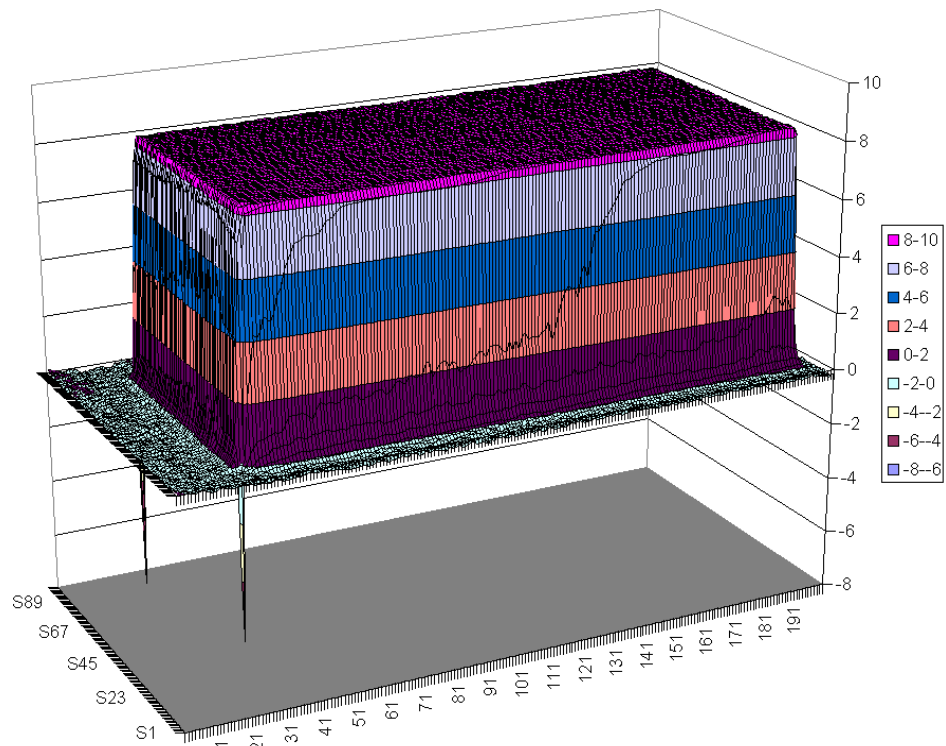
**Figure 7-15: Combined Gauge Block Output using Nearest Vector Orientation Method based on digital camera image. Data 'trough' visible at 'top-left' corner due to scan region misalignment.**

As discussed in Section 4.3, compensation algorithms can be applied to correct the camera image. In order to show what can be achieved when the camera image is not distorted 'manual' calibration points were defined to correlate the pixel coordinates of the corners of the gauge plate in the image space with their locations in the scan files.

When these manual calibration points are used with the webcam image, the resulting scan regions are closely aligned with the position of the gauge plate (Figure 7-16) and the combined scan errors are largely reduced (as shown in Figure 7-17). Remaining errors are due to registration errors (i.e. the problem of precisely defining the corner position of the gauge plate when the scan file includes data distortions). Similar (albeit less dramatic) improvement can be seen in the alignment of the digital camera image when manually defined calibration points are applied.



**Figure 7-16: shows corrected scan region map with alignment defined by manually selected calibration points corresponding to the corners of the gauge plate**

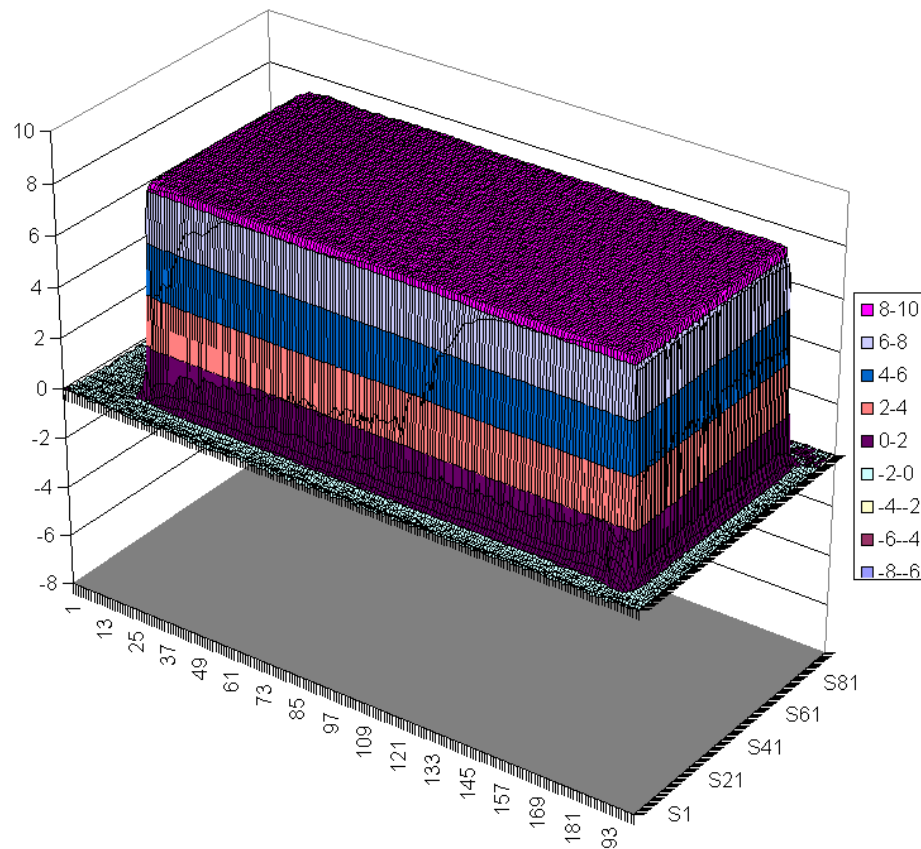


**Figure 7-17) Output from Nearest Vector Orientation Method using webcam and manually selected calibration points (see Figure 7-16a) shows much improvement scan region alignment over that shown in Figure 7-14. Remaining data errors are the result of differences in scan registration.**

The improvement in the quality of the scans suggests that the calibration across the image is non-linear. The poor quality of the webcam optics results in poor alignment, even when the calibration markers are placed as close as possible to the object (to minimise the effects of lens distortion). The digital camera image displays considerably less distortion; the image is better focused overall resulting in the scan regions being more closely aligned with the correct position and no significant distortions can be seen (Figure 7-18).

The edge parallax effect also causes edges closer to the camera to appear shifted away from their true position by a greater extent than those edges further from the camera. This effect increases with distance from the centre of the image. This will also contribute in causing displacement of the image scan regions from their 'correct' position relative to the scan edge and requires some compensation (as discussed in Section 4.3.2).





**Figure 7-18 Combined output by vector orientation method on gauge plate using digital camera. Data spike and bow wave effects have been almost completely eliminated by applying manual calibration.**

Because the gauge plate has a simple geometry with constant known height it is possible to calculate a numerical error value for each scan method compared to a ‘ground truth’ for this object. This is not generally possible for more complex objects with an irregular profile. All values should ideally be one of two height measurements, representing either the top surface of the block or the base height of the surrounding scan bed.

Although the dimensions of the block are known, the addition of a coating of the Flawfinder Developer spray prior to the scan causes a small change in the recorded scan readings. The true height of the gauge block is measured at 8mm with a micrometer screw gauge, but the average top surface height as measured by the Matsushita laser was 8.38mm, (suggesting that the thickness of the spray coating added ~0.4mm to each side of the block).

An average height of the surrounding scan bed area was calculated (excluding those areas where scan errors are known to be present) then all the scan readings were adjusted relative

to the average base height. An ‘ideal’ output scan representing the ‘ground truth’ object was created, where all values were either 8.38mm (representing the top surface of the block) or 0.0mm (representing the surrounding scan bed area). The original 90° scan (which represented the dominant single scan orientation) was used as a guide to locate the position and orientation of the ‘ideal’ block within the scan-space. All scan points above a threshold height of 6mm were assigned to the object. Using a mean square error calculation, the following error values were calculated for each scan.

Scan (20,000 scan points)	Root Mean Square Error (mm)
Worst Single Orientation (0° scan)	2.129
Best Single Orientation (90° scan)	0.969
Scan Region Orientation Method	0.652
Nearest Vector Orientation Method	0.518

**Figure 7-19: Table of Root Mean Square Errors for the different methods compared to a standard ‘ground truth’ scan. The combined methods show much less error than single orientation scans**

As expected, the error level in the 0° scan is significantly worse than the dominant orientation. It can be seen that both the new methods produce an improvement in the average error level within the scan. Despite the magnitude of the mean errors it can be seen that both the SRO and NVO methods have a lower mean square error than the standard ‘default’ direction and the NVO method is somewhat better than the SRO method, which supports the qualitative assessment of the data.

Some of the RMS error in the scan data can be accounted for by hysteresis across the vertical edges of the block. Even if an edge is encountered at the ‘perfect’ orientation there is some ‘fall-off distance’ between the top and bottom of the edge (see Figure 2-18). Because the edges of the object do not align perfectly with the scan sample points some values around the perimeter represent points registered where the sampled position is part-way up the vertical edges. These points have intermediate values between the base and top surface height that were rounded up or down in the ground-truth comparison data, leading to a number of points with significant error values around the perimeter of the object. Also the block is not aligned perfectly with the axes of the scan, again resulting in the scan line ‘climbing’ the side of the block and registering as a series of increasing edge height over consecutive scan points. This contributes to the calculated error level in comparing to the ‘ideal’ ground truth scan.

### 7.4.2 Summary of Image Distortion Problems

The results from the domino and gauge plate scans show that the camera calibration is a major factor in the quality of the final scan output. Investigation of the alignment errors that occurred when using the webcam to provide the calibration image showed that the discrepancies may be caused by two major factors: firstly the positions of edges in the camera image are offset from the ‘true’ position by the parallax effect on the observed image edges. This is especially prevalent in this image due to the length (in the x-axis) of the gauge plate relative to the camera and object height. The edge parallax effect also causes edges closer to the camera (i.e. in the z-axis) to appear shifted away from their true position by a greater extent than those edges further from the camera. This will also contribute in causing displacement of the image scan regions from their ‘correct’ position relative to the scan edge and requires some calibration as discussed in Section 4.3.2.

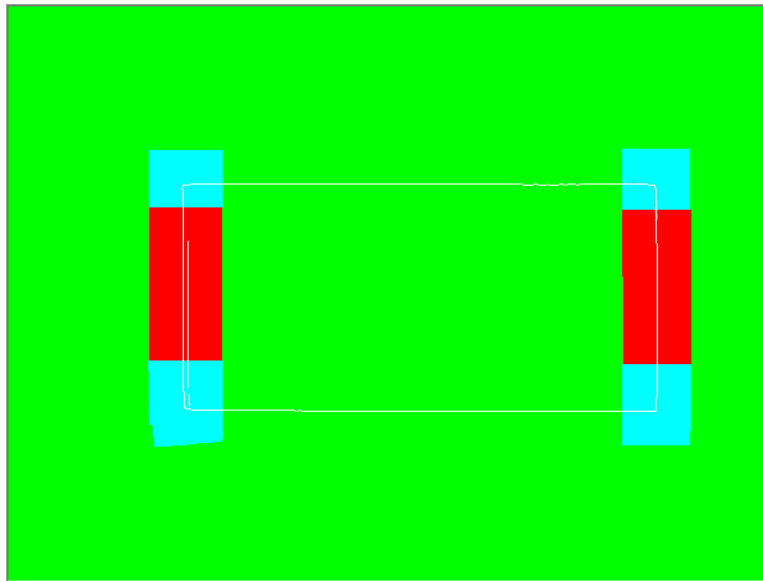
Secondly, the poor image quality of the webcam shows the effect of radial distortions may have in causing a poor calibration between the observed marker positions in image space and their actual positions in scan space.

Even if the image processing is able to determine ‘perfect’ scan regions based on the available camera image information, the selection of scan orientation regions will be incorrect unless the output map can be related to the actual object position. The quality of the camera optics and the reliability of the calibration mechanism, are therefore important parts of the full system. The poor quality of the webcam optics renders the output from the combined partial scan to be almost as bad as a single scan results.

A similar condition may occur if the laser scanner is not registered correctly (see Section 4.3). However this is not examined here as all the pre-scans were registered as closely as possible by the selection of an initial scan point. It should be noted that the correct alignment of the laser is of the greatest importance when multiple partial scans must be integrated. Any ‘pirouette’ effect of the emitted laser beam as it is rotated will cause differences in calculated height and transposed positions in the x and y axes between the different rotations, leading to difficulty in combining the scans from different orientations.

## 7.5 Overlapping Regions

Situations where scan regions overlap only occur with the 'Scan Region Orientation' method. This method works well in areas of the scan where a single orientation is dominant. However, the method breaks down where regions of significantly different orientations overlap. In Figure 7-20 the areas corresponding to the  $0^\circ$  scan regions are shown in red and the  $90^\circ$  scan regions are indicated in green. The positions of scan region overlap are indicated by the blue areas.



**Figure 7-20 Gauge Plate (digital camera) Scan Region Orientation Region Map showing region overlaps (in blue).**

Although the scan regions detected in Figure 7-20 correspond well to the edge positions of the gauge plate, the overlap of regions representing significantly different orientations causes localised problems, as shown in the composite scan output view Figure 7-21. Figure 7-21a) shows the view from the 'bottom left' corner (relative to the image shown in Figure 7-20) image and 7-21b) shows the view from the bottom right.

The spike and trough errors have been successfully removed where scan regions do not overlap. The overlapping regions at corners show errors due to the simple averaging process used to determine values at these locations. This is unavoidable when the scan region orientation method is applied. The particular combinations of bow waves and spikes at the corners give different output based on the orientations of the original scan orientation files. Figure 7-22 relates the errors A, B, C and D using a view from above the object.

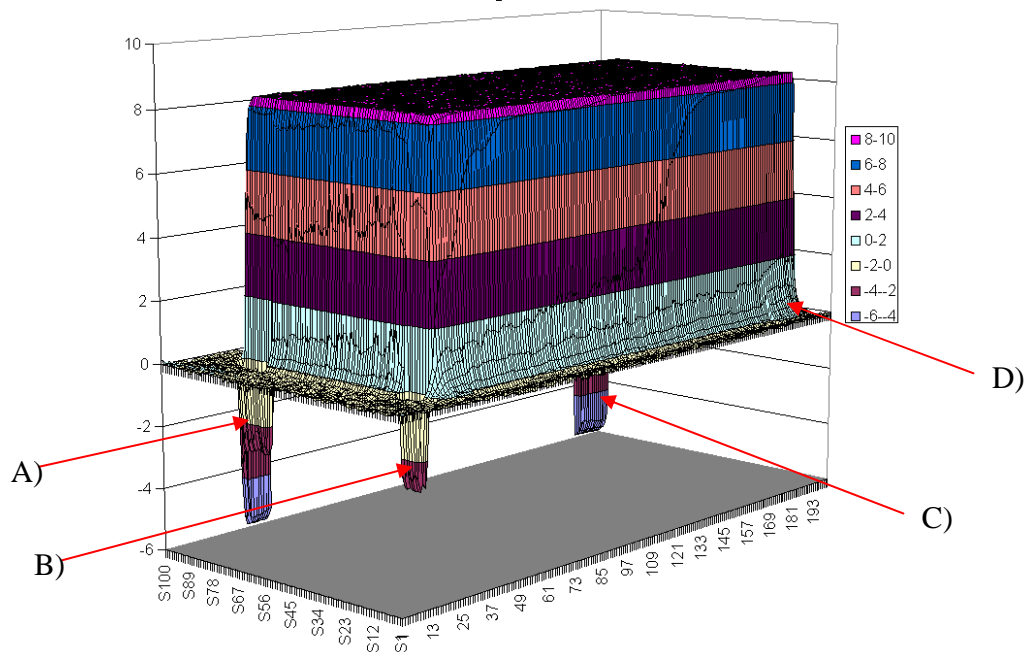


Figure 7-21 a) View from 'bottom left' of image –combined output using scan region method – shows good removal of errors where regions do not overlap. A,B,C & D mark the errors caused by overlaps

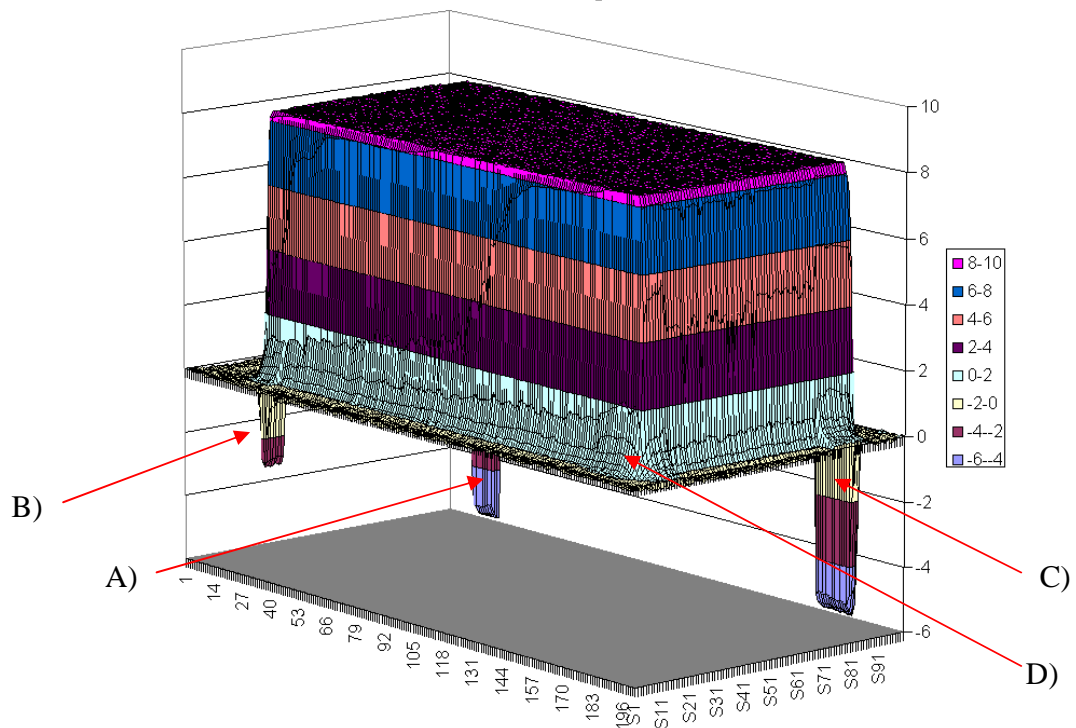
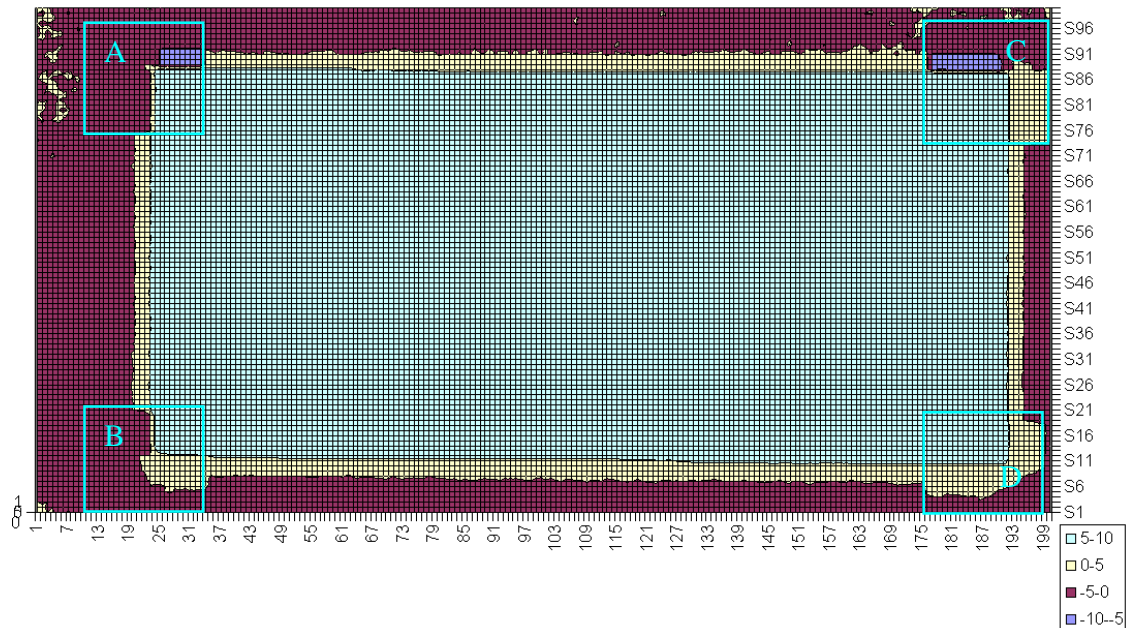


Figure 7-21b) As above, view from bottom right of image – remaining troughs and bow waves are the result of the overlapping scan areas. A,B,C and D mark the same errors as above



**Figure 7-22 shows the view from above the gauge plate combined scan using the scan regions.**

The marked combinations of the combined scan areas are indicated. In each case, the degree of the error in the overlapping regions is reduced compared to that of the single scan where the error occurs, but worse than if the correct selection of orientation data is selected using the Nearest Vector Orientation method.

Where the overlapping regions represent orientations that are significantly different then averaging the data often fails to provide a 'good' output value (i.e. the mean value is not representative of the 'true' edge position).

- A) In this overlap region a combination of 'trough' spikes acquired from both the contributing scans ( $0^\circ$  and  $90^\circ$ ) appear on both sides of the corner.
- B) Shows trough spikes on the left edge (from the  $90^\circ$  scan) and bow wave on the bottom edge (from the  $0^\circ$  scan).
- C) shows trough spikes on the top edge (from the  $0^\circ$  scan) and bow wave on the right edge (from the  $90^\circ$  scan).
- D) shows bow waves on both sides of the corner, acquired from both scans.

Overlapping regions of orientation may produce good results in some cases, especially where the overlaps between scan regions represent similar orientations (as shown in Section 7.6.1); however the scenarios which favour the Scan Region Orientation method are limited. Overlaps of different orientations often occur within the area of the object's internal geometry due to the required minimum region width. The averaging of such areas tends to produce a poor quality output (unless all orientations produce very similar data) and in most cases the results are inferior to those produced by the Nearest Vector Orientation Method.

The Scan Region Orientation method is significantly faster than the Nearest Vector Orientation method. The two methods could be used in conjunction to reduce the overall time required to calculate the scan region map with the SRO method being used to provide a base region map and then the NVO method applied only to those areas where regions of different orientations overlap. The amount of time saved would vary depending on the complexity of the object under consideration (i.e. proportional to the amount of overlap in the image).

## **7.6 Edge Curves and Corners**

Curved edges represent a challenge for the proposed methods, as they rely on the development of straight line segments of a minimum length. The ability of the methods to ‘track’ around a curve is important as most objects will include at least some detail that does not consist of entirely of straight lines.

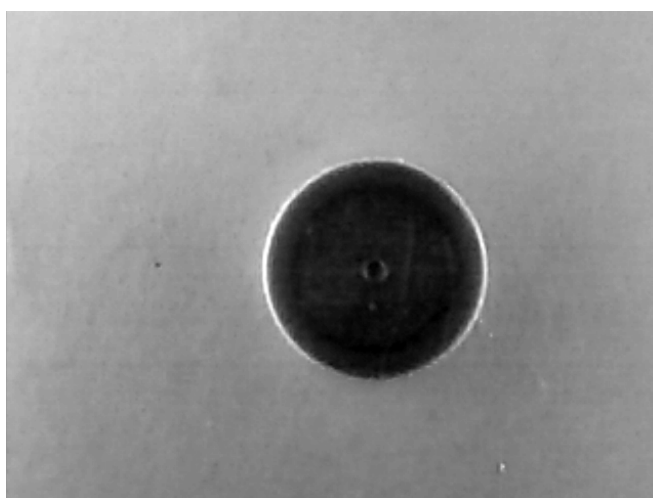
The magnitude of curvature will affect the ability of the method to track the curve closely, although the line segment length is a ‘tuneable’ parameter of the vectorisation process there is a practical minimum limit on the length related to the scale of detail in the object and the presence of noise in the image. There is usually some ‘tolerance’ in the scan region map for the selection of a scan orientation which is close to the actual line orientation (as discussed in Chapter 4.1). This allows a smooth curve to be approximated by a series of straight line segments which follow the general direction of the arc without precisely tracking every pixel. The limits at which the orientation tolerance breaks down depend on the curvature of the edge and the proximity of other scan regions with different orientations. Corners represent the greatest challenge. This is shown in the following subsections.

### **7.6.1 Low Curvature (Bottle Cap)**

The simple method of averaging the data selected by more than one scan region orientation produces fairly good results under the limited conditions where one orientation region overlaps another region of close orientation and has shown the ability of the technique to select orientations representing a smooth transition around a curved edge. For example, if Region A, representing a  $0^\circ$  orientation scan overlaps with Region B representing an  $18^\circ$  orientation scan then the data within the overlap will provide a reasonable ‘blend’ between the two areas. Similarly Region C, (representing a  $36^\circ$  orientation scan) may overlap with region B without obvious distortion, although if C overlaps A then there may be a significant disagreement between data values for scan point within this region of overlap.



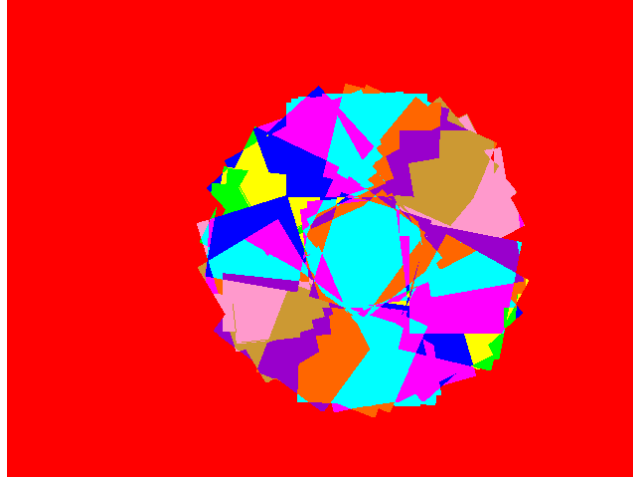
This is shown in the example of the bottle cap (as shown in Figure 7-23). The object shows a profile with consistent low curvature around the perimeter. Any single orientation scan over this object will cause similar errors (although shifted around the circumference of the object, relative to the scan orientation). By detecting the edges around the object's circumference it is possible to select the corrected partial scan orientations to produce an accurate composite output file. For reference, the cap is approximately 40mm in diameter and 9mm high. It is made of a matte plastic and side-lit to highlight the edges.



**Figure 7-23: 640x480 greyscale image of the bottle cap used as a basis for edge detection, vectorisation and region building.**

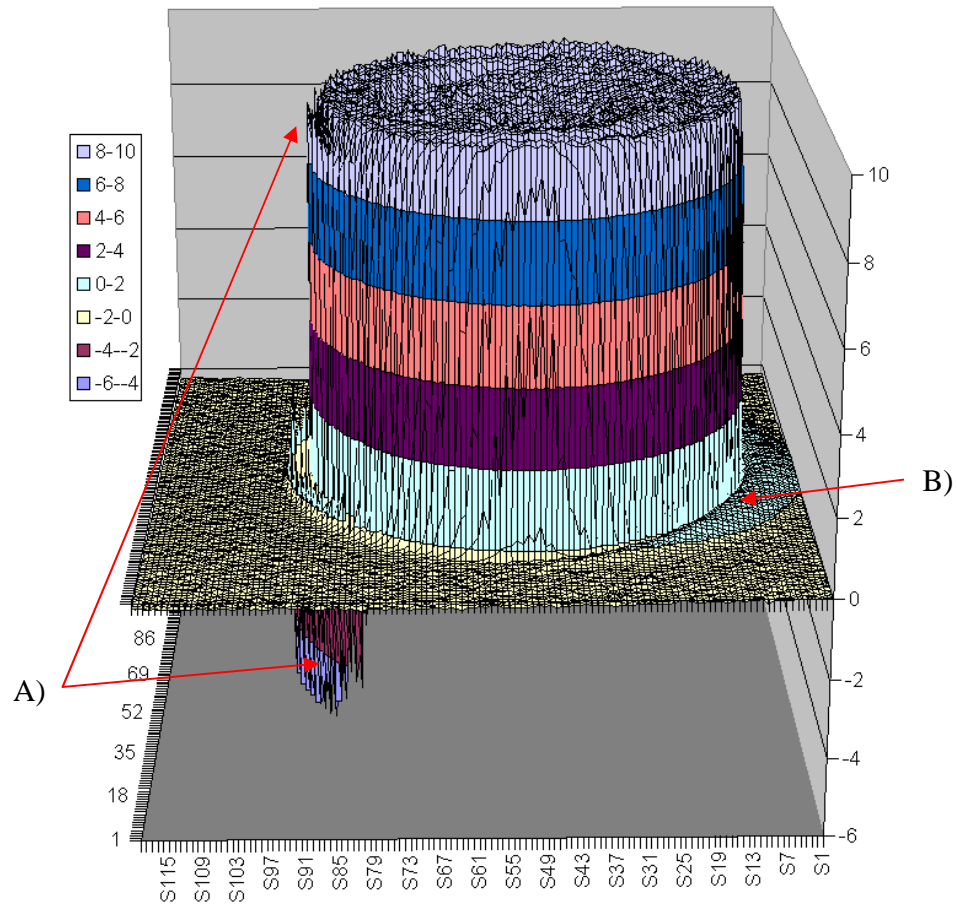
The results of the Scan Region Orientation method show a large amount of overlap between the regions of different orientations however, for the most part, these region overlaps do not strongly disagree about the orientation angle of the data, as they follow a 'smooth' progression around a curve, so they are generally only one 'orientation set' of data apart. The output map shown in Figure 7-24 took 2.05 seconds to produce based on the above vectorisation map.

Overlaps in the central area of the cap would be more troublesome, however the scan data in that area does not ‘disagree’ significantly between the different orientations; as the edges detected there represent only a slight change in the surface (due to the moulding extrusion process) it does not make too much difference which orientation is selected within that area.



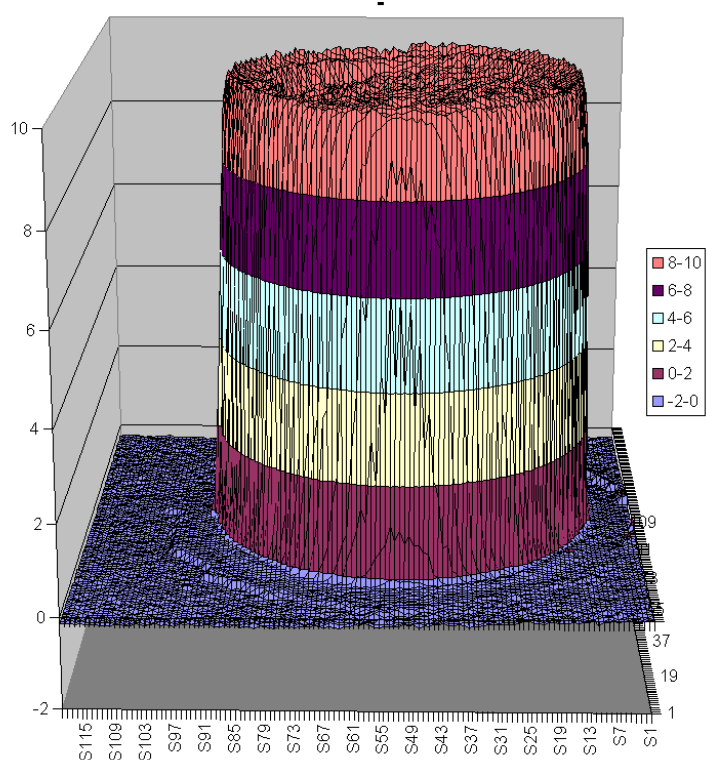
**Figure 7-24: Output by Scan Region showing overlapping regions over surface. Because of the limited colour palette of the image it is difficult to show the extent of the combinations of overlapping areas**

The 0° scan was determined to be the dominant direction because slightly more vectors were detected within that orientation. Because the object is circular, in reality there is not any particular orientation that will give significantly better results, however the scan presented in Figure 7-25 is used as a reference to which the combined scans will be (qualitatively) compared.

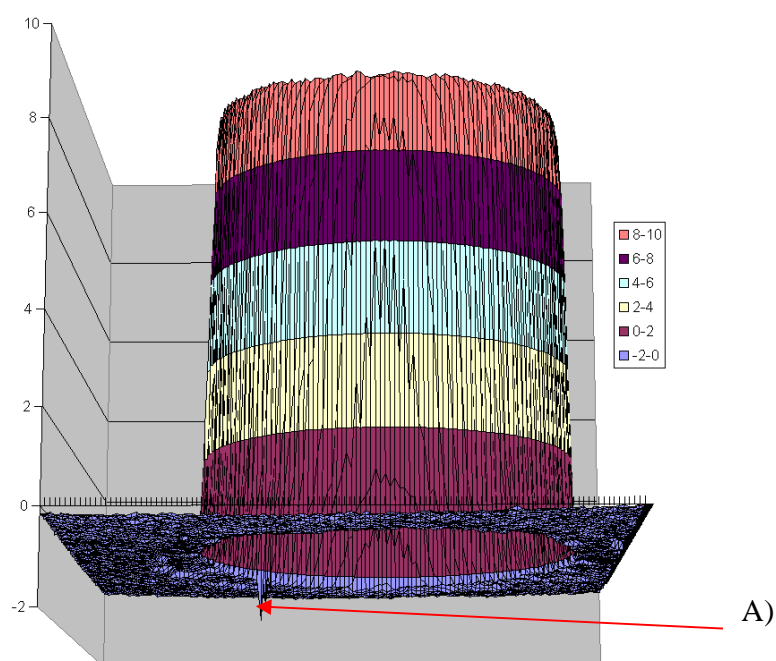


**Figure 7-25: ‘Best’ Single Scan orientation (0 degrees) shows spike / troughs (A) and bow wave (B) around edges (view from LH side with respect to camera image. i.e. y-axis of scan is x-axis of this diagram in order that both bow wave and spikes are visible).**

Figure 7-26a) and 7-26b) show the output using the scan region orientation (SRO) method. In this scenario SRO is quite effective and the removal of the spikes and bow wave effect from around the base of the cap can clearly be seen. The apparent shapes of the areas of overlapping regions can be observed on the base due to poor height registration between different scans.

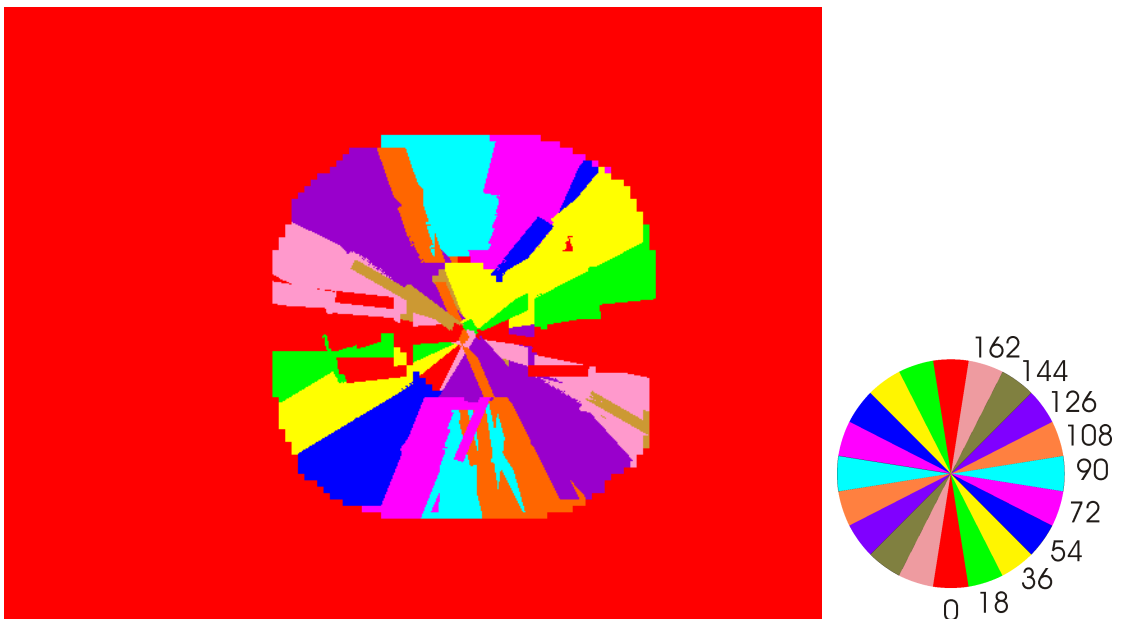


**Figure 7-26a) Output by Scan Region Orientation – (same view as 7-25). Troughs / spikes are mostly gone and bow wave removed.**



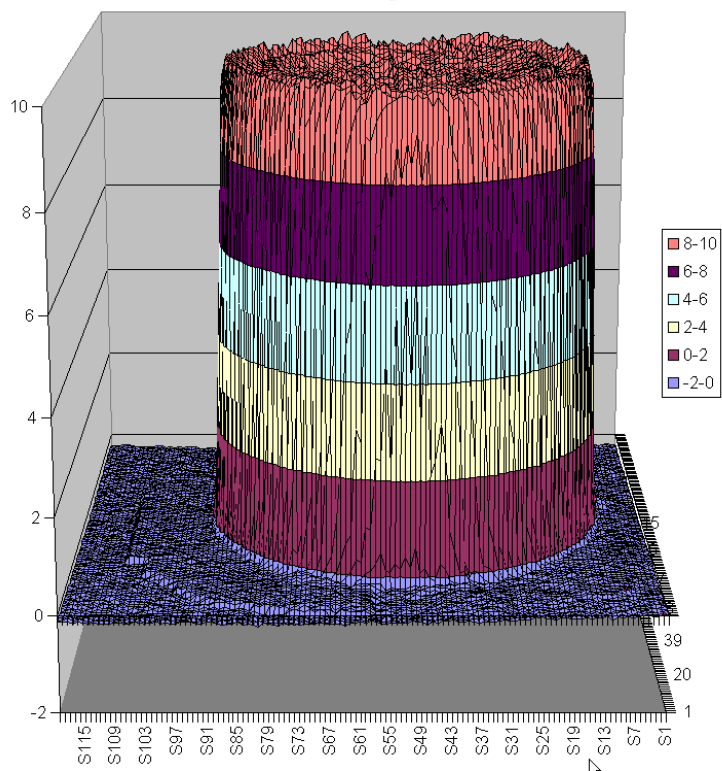
**Figure 7-26b) Output by SRO (view ‘underneath’) – shows a slight ‘missed’ trough A) using the scan region method due to the region overlap averaging**

The scan region map produced by the Nearest Vector Orientation (NVO) method shows a reasonably close correspondence between the output file and the ‘ideal’ orientation (from the circular colour wheel), as shown in Figure 7-27. The regions extend across the edges quite cleanly. Although there are some vectors that cause scan regions to be created at slightly different angles from the expected direction, in general these are not at an orientation too different to the expected orientation at any position. This scan region map was used to produce the output shown in Figure 7-28 in 10.15 seconds. This output takes 5 times as long to produce as the SRO method, but this is still much shorter than the actual scan process.

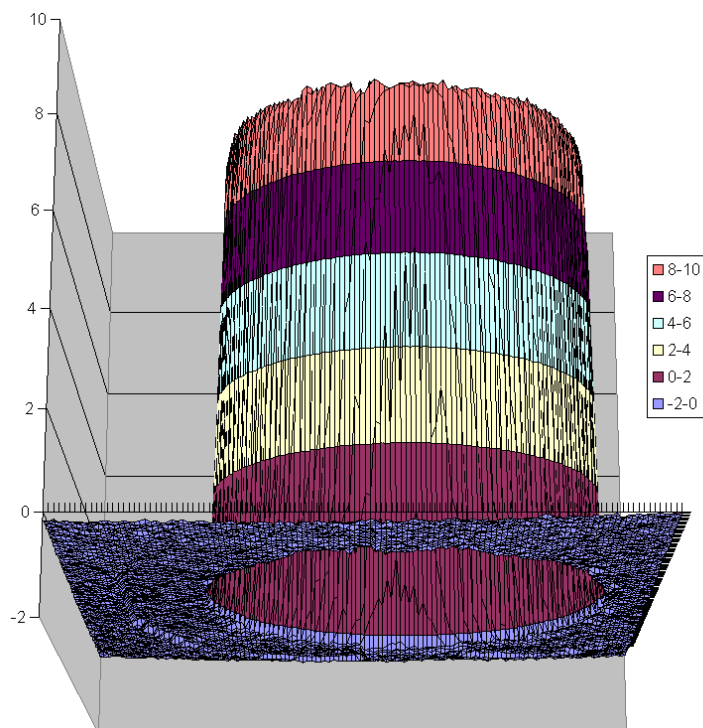


**Figure 7-27: Output by Nearest Vector Orientation corresponds closely with the ‘ideal’ vector orientation wheel (right)**

Figure 7-28a) and 7-28b) show the output from the NVO method. Data spikes and the bow wave effect are effectively removed from around the base of the cap. The extent of the areas of orientation regions can again be observed on the base due to poor height registration between different scans. This also results in the somewhat ‘choppy’ appearance of the surface of the cap itself whereas the previous (SRO) method disguises this ‘choppy’ edge by averaging the overlapping data. (Although the height variations are actually very small, they appear larger due to the z-scaling of the graph).



**Figure 7-28a) Scan output by nearest vector orientation shows good removal of occlusion and secondary reflection effects.**



**Figure 7-28b) Scan Orientation by Nearest Vector Orientation from underneath – shows all trough spikes removed from around circumference**

The results from both partial scan region methods are much better than a single orientation scan. The SRO method manages to control the occlusion and reflection errors effectively without requiring a scan region width that causes averaging of conflicting scan region orientations. Where regions do overlap they tend to vary in the reported data value by only a small amount because the overlap is ‘progressive’ around the circumference.

The NVO method is successful in removing all the occlusion errors, but the poor registration in the z-axis results in a rough appearance to the scan output which is less apparent in the scan region method because of the averaging between overlapping scan orientations. This problem with the different base height is a result of internal processing within the scan generation (Scan3D) program, which attempts to adjust the base height of each scan automatically, rather than a fault of the test program. Either method can therefore be applied successfully to regions of low curvature with no ‘intruding’ scan regions of a conflicting orientation.

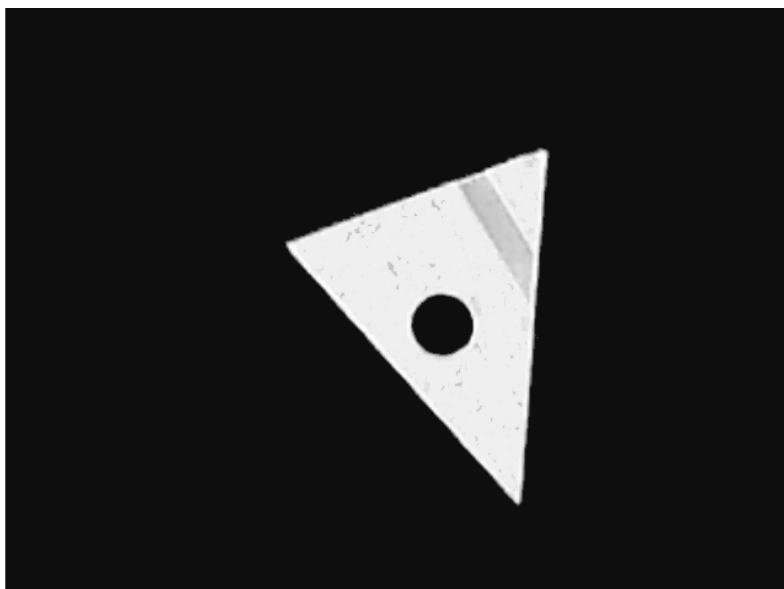
### **7.6.2 High Curvature (Triangle Block)**

The triangular metal block (shown in Figure 7-29) provides an example of a simple machined object with external straight, vertical edges (10mm high) the two longer sides are approx 35mm long. It contains a linear engraving across the surface and a through-hole with a high curvature (10mm diameter) that is expected to cause problems, because the laser sensor will have difficulty ‘seeing’ the bottom of the hole at every sensor orientation.

The object was sprayed with Flawfinder Developer spray to reduce specular reflections from the metallic surface. However this process considerably reduced the contrast of the object’s surface. In order to bring up the detail the image intensity was equalised<sup>1</sup> prior to edge detection. The calibration edge detection and vectorisation process was followed as for the previous objects and is not detailed here.

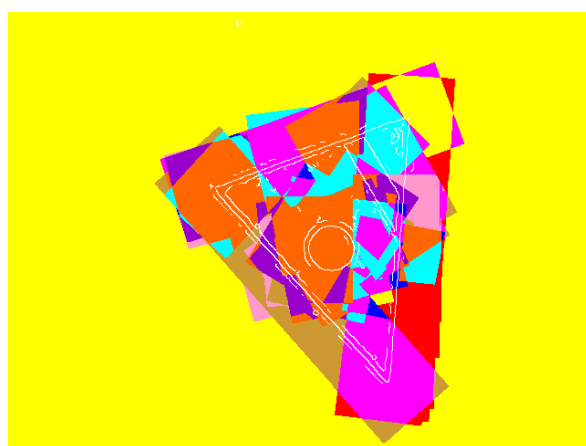
---

<sup>1</sup> Image equalisation is a histogram modeling technique that provides a means of modifying the dynamic range and contrast of an image. Histogram equalization employs a monotonic, non-linear mapping which re-assigns the intensity values of pixels in the input image such that the output image contains a uniform distribution of intensities.

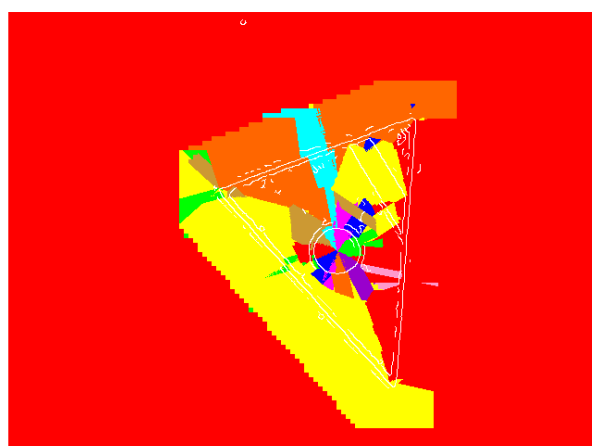


**Figure 7-29 shows the greyscale image of the triangle block**

Scan region building for the triangle object using the SRO method is somewhat more complicated than for the bottle cap because there are so many small regions overlapping with conflicting orientations. The required minimum width of each region extends over much of the surface, as shown in Figure 7-30a). Due to the limited colour palette available it is not possible to show the overlaps clearly, but it can be seen that all regions overlap at least two others.



a)



b)

**Figure 7-30: a) Output Map by Scan Region Orientation (left) for triangle shows a large amount of overlap between regions – region map calculation took 1.99 seconds for this output. b) Output Map by Nearest Vector Orientation shows better orientation matching – region map calculation took 10.26 seconds for this output.**



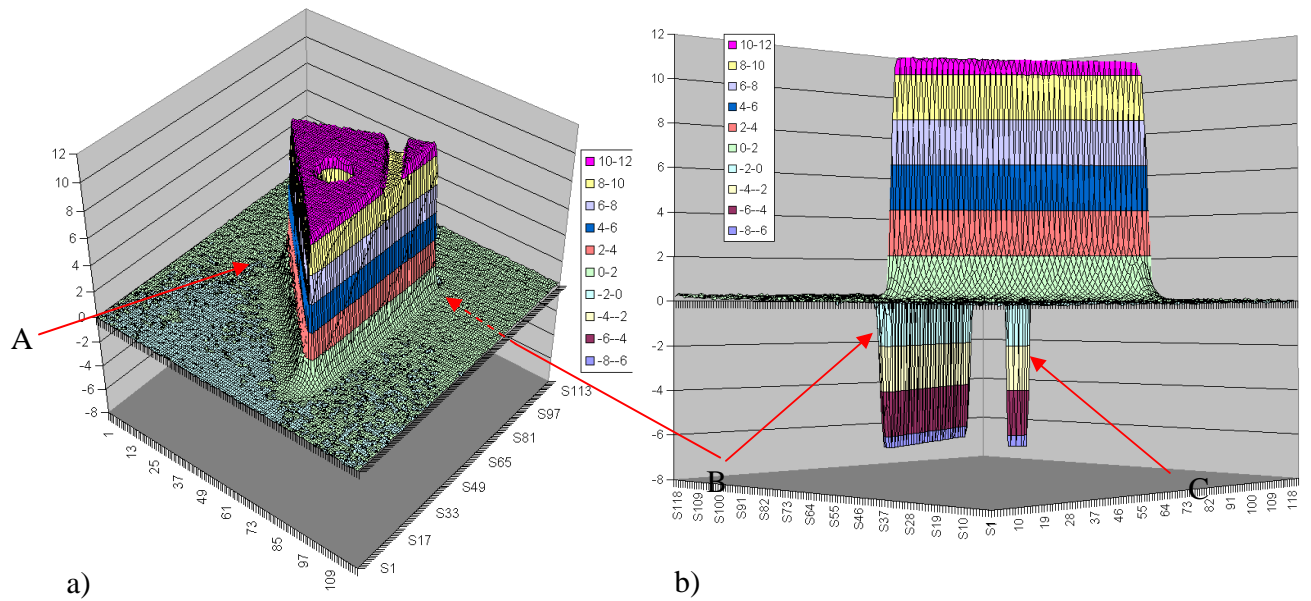
The nearest vector method (Fig.7-30b) does not suffer from this overlap issue and therefore presents a much clearer picture of the object's edge orientations. The ability of the NVO method to resolve small features such as the through hole is shown by the 'wedges' representing different orientations at the location of the through hole in the image. Calculation of the scan output map took approximately 5 times longer (~10secs) than the scan region orientation method (although this is still far less than the time taken for a single scan over the object).

One point of interest from the two methods is that the calculation of the dominant scan direction results in the selection of a different base orientation between the two scan region techniques. The Scan Region Method calculates the base scan on the area of each orientation after the regions at each orientation have been merged, whereas the Nearest Vector Orientation method calculates the base scan calculated by the sum of vector lengths per orientation.

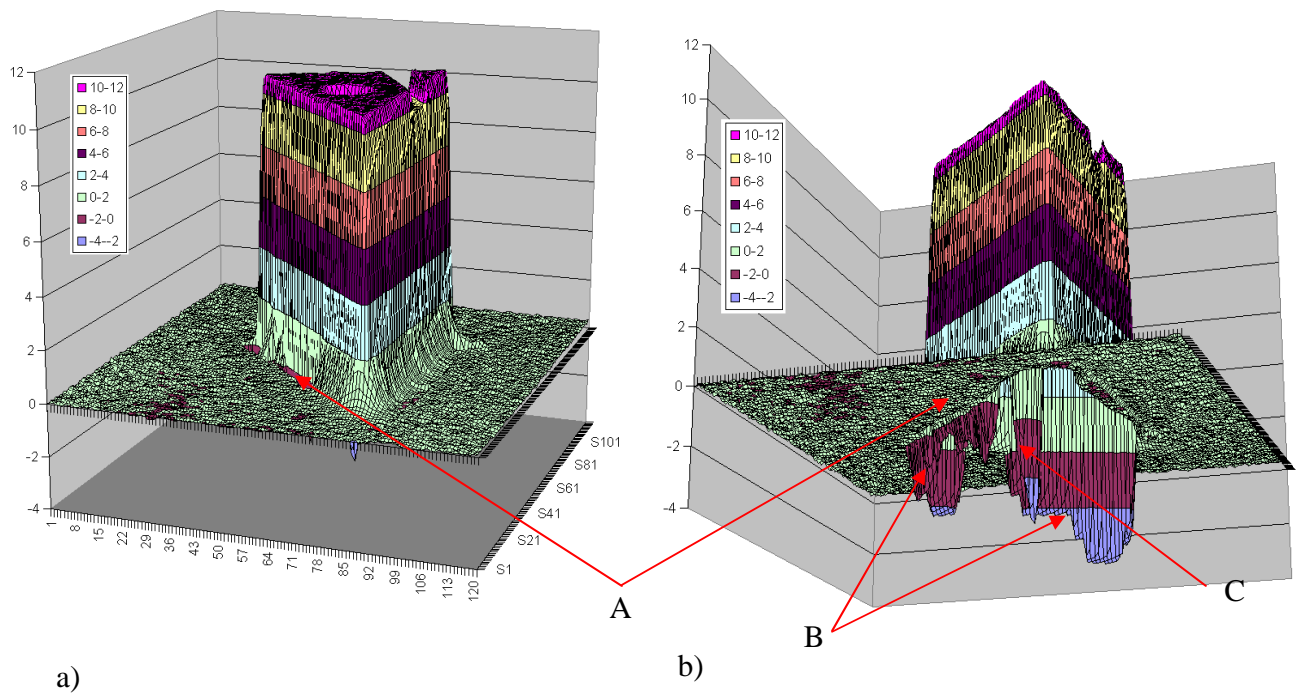
For the nearest vector orientation method the 0° scan is the dominant orientation (this orientation is shown for comparison in Figure 7-31). This looks reasonably clean from the view (a), although some bow wave 'buttressing' can be seen along the edge marked A) at approximately 30° to the orientation of the scanner. However from the side view (b) it can be seen that there is a large 'trough' along the edge running within the 108° area and at the position of the through-hole.

Applying the SRO method demonstrates the problems associated with many overlapping regions of different orientation. There are so many overlaps that the averaging of the data results in some errors being introduced at almost every position where there is a significant 'disagreement' between output values for different orientations. In many points there are several scan regions of incorrect alignment overlapping a single 'correct' orientation resulting in the magnitude of the errors significantly outweighing the correct data.

In the best case, errors that would be correctly removed by one single correct scan region orientation are instead averaged, causing them to be reduced but not removed. Generally however, errors that were not present in the 'best' single scan orientation are introduced into the combined scan by the averaging process. These errors are apparent in Figure 7-32.

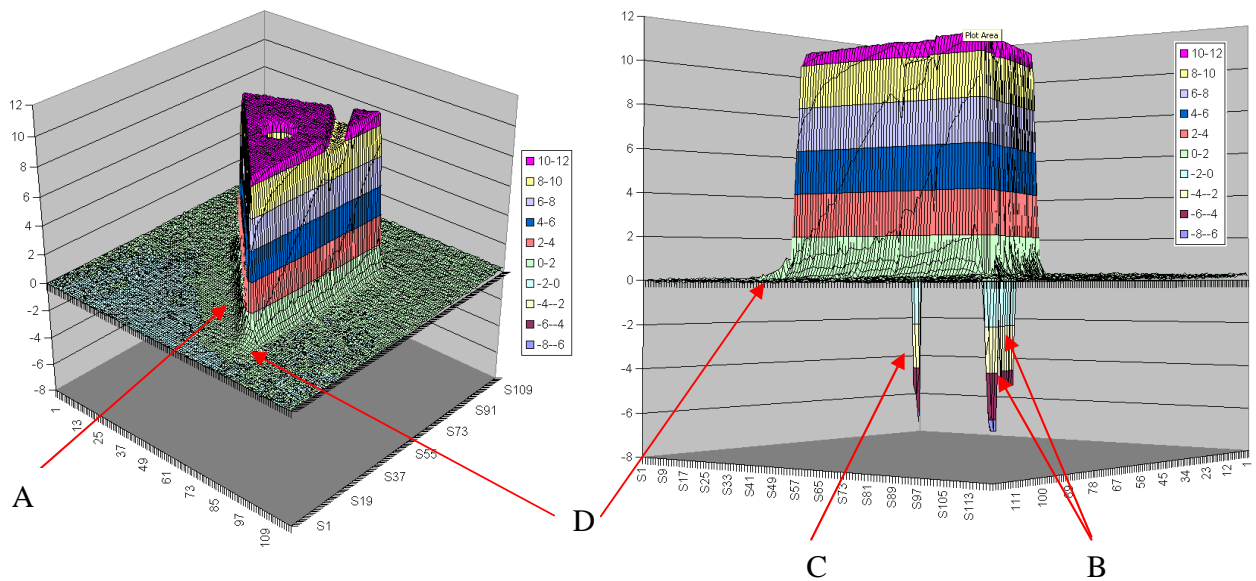


**Figure 7-31 a) Best Single Orientation Output (0 deg) for triangle scan A) marks bow wave ‘buttrussing’ of edge at 30° angle to scanner orientation. However this scan looks poor from side view (right) – shows large trough (B) along edge (top edge in image) and at position of through-hole (C)**



**Figure 7-32: Triangle Widget Combined Output by Scan Region (based on 36° scan). Left image shows scan of object from side. Image b) shows scan from underneath. Many errors (A and B) are visible along the edges due to averaging of overlapping regions. C indicates position of through-hole, where some error is unavoidable due to sensor occlusion at all orientations.**

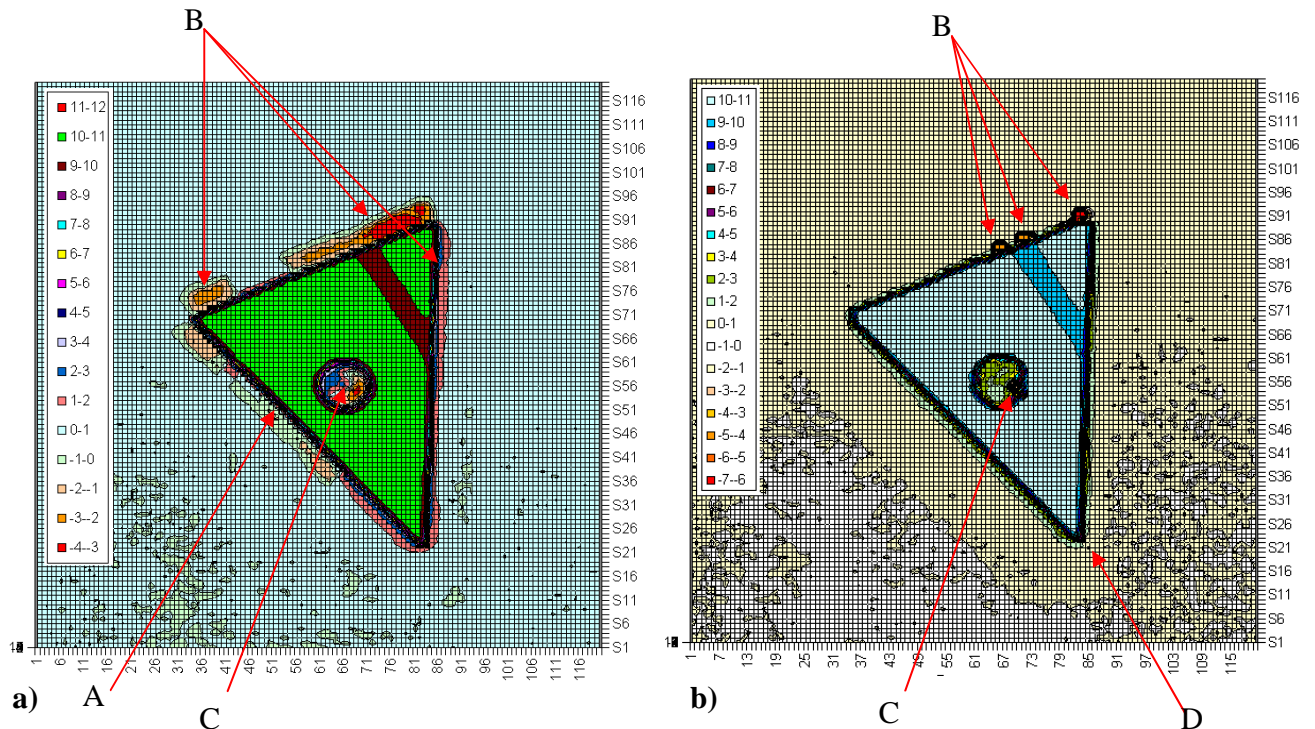
The nearest vector orientation method produces a combined scan that appears much better than the scan region method. There are some errors in the resulting file where the scan regions are not well-aligned with the original file probably due to calibration errors, but the result shows much reduction in errors compared to the single 'best' orientation scan. The result of applying the NVO method to the scan data is shown in Figure 7-33.



**Figure 7-33: Combined Output by Nearest Vector Orientation based on 0 degree scan from above shows no significant errors and reduced bow wave along edge (A). The 'trough' along the edge from the default direction scan (labelled 'B' in Figure 7-32) is reduced to 3 narrow data spikes. The 'through-hole' error (C) is reduced in size to a narrow trough-spike. A slight bow wave effect remains, visible at D) although overall reduced in size from default scan direction.**

The NVO method proves much more successful than the SRO method. Despite taking several times longer to determine the scan orientations this is negligible compared to the time for a single scan of the triangle object. (Even at a relatively coarse scan resolution of 0.5mm the sensor takes approximately 10 minutes to traverse the scan area).

Figure 7-34 shows a view from directly above the resulting combined output from each method. Errors around the perimeter of the object can clearly be seen in both scans, although the quantity and magnitude of errors in the NVO scan (b) are significantly less than the SRO scan (a). The SRO errors (A & B) are associated with the averaging of different region orientations.

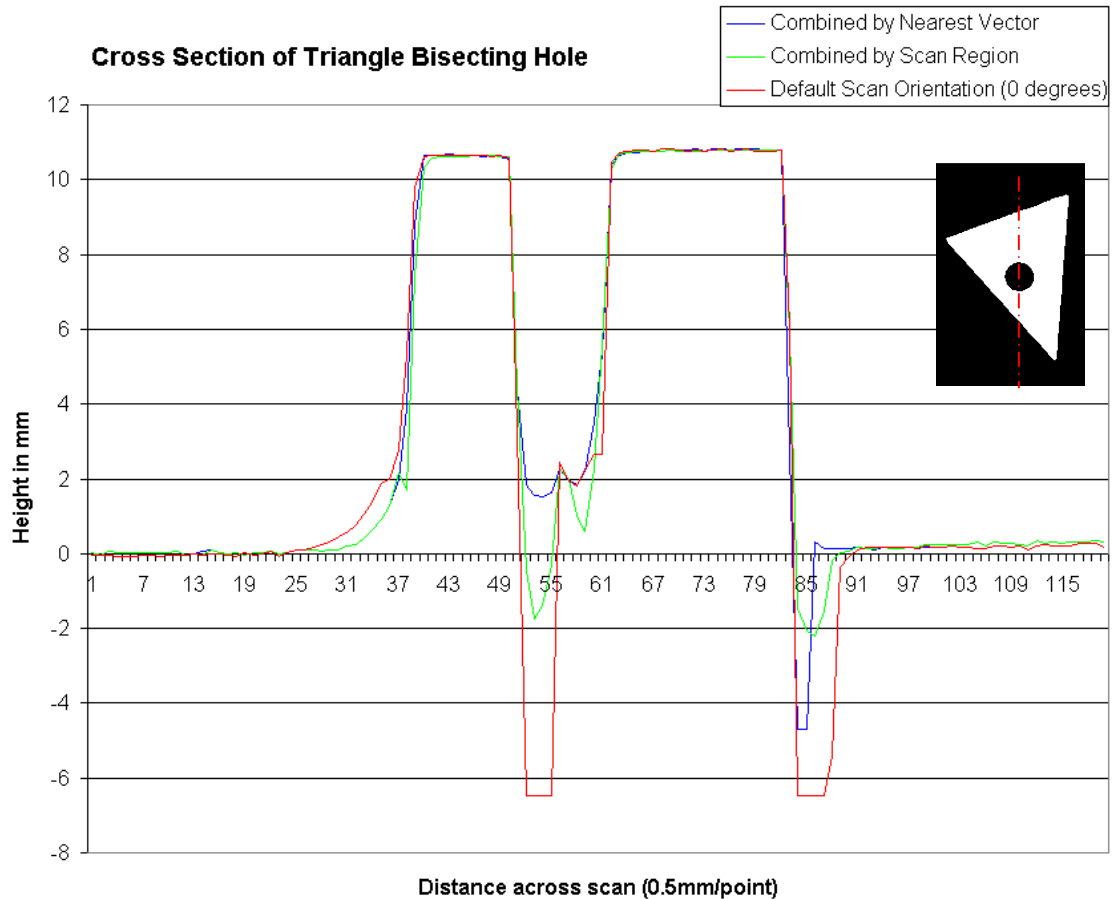


**Figure 7-34 a) shows top-down view of Scan Region Orientation output showing results of overlapping region averaging. b) shows same view of output by Nearest Vector Orientation. The location of the 3 spikes (B) along the top edge in proximity to the junctions of different scan regions. Note labels correspond to the same features described in Figures 7-32 & 7-33**

The remaining errors in the NVO scan are primarily associated with localised junctions of regions of different orientation where an incorrect value has been selected. There are several possible reasons for the incorrect data. These issues are related to the selection of data at corners (i.e. positions of extreme curvature) and are covered in Section 7.6.3. The errors at (B) in the NVO scan are most likely to be due to poor calibration and therefore although the region map indicates the correct orientation a conflicting orientation is selected.

The area within the through-hole (C) generates errors due to sensor occlusion and secondary reflections at all scan orientations, as expected. The slight offset of this error away from the centre of the through hole also suggests that the calibration has caused slight misalignment in the data selection. Although the NVO method does help to reduce the magnitude of this error it is almost accidental because there is no point at which the sensor reports the correct base height and values have been selected more from the ‘secondary reflection’ than the ‘spike’ region of the hole.

A cross section through the default and combined scan profiles (Figure 7-35) bisecting the hole shows this more clearly.



**Figure 7-35 shows profile of scan through hole in triangle. The data representing the internal area of the hole is composed of a reflection ‘bow wave’ plus occlusion spike, with no correct data points. Inset shows line of section through object.**

In summary, the SRO method is not able to handle edges representing high curvature very well, due to the required minimum region width causing a large degree of overlap between different orientations. The NVO method does not have a set region width and can therefore cope with areas of high curvature, however the calibration of the scan region map and the scan space is critical in these areas representing small features or the scan will be misaligned, resulting in the incorrect orientation being selected at the region boundaries.

### 7.6.3 Corners

Corners are locations in the image space where the change in orientation between intersecting (or adjoining) vectors exceeds the orientation tolerance<sup>2</sup> of the sensor by an angle sufficient to ‘switch’ between disjoint region orientations (i.e. one or more orientations around the ‘orientation wheel’ are omitted). The simplest example being 90° angles such as those on the external corners of the gauge plate.

It was shown in Section 7.4 that the Scan Region Orientation method produces poor results in such situations, however the Nearest Vector Method can also generate poor results because of poor calibration or poor results from the edge detection / vectorisation stages.

At corners the correct selection of edge detection and calibration are critical in selecting the correct region orientation for each scan point. If the corner position is found correctly then the scan region should precisely bisect the corner, and errors will be avoided. However if the edge detection or vectorisation finds the corner in the ‘wrong’ position (relative to the position of the corner in the scan space) or the calibration at that position in the image is inaccurate (see Section 7.4) then an incorrect orientation will be selected close to the corner. This can often cause a localised data spike.

The alignment of the scan region map with the object is critical at these boundaries between non-contiguous orientations as there is no margin for error at these locations. Correct localisation of junctions such as corners is therefore important in the edge detection / vectorisation process. The use of high Gaussian smoothing values to reduce noise in the original image will also have the unwanted effect of reducing the accuracy to which corners are detected. Further ‘corroborative’ methods of corner detection could be employed to provide a ‘second opinion’ on corner positions at the image detection stage.

One solution to this problem could be to apply Wong’s method in regions immediately surrounding corner features. Whilst this would require the collection of a full set of orientation data this would only be required for a relatively small total area of the scan.

---

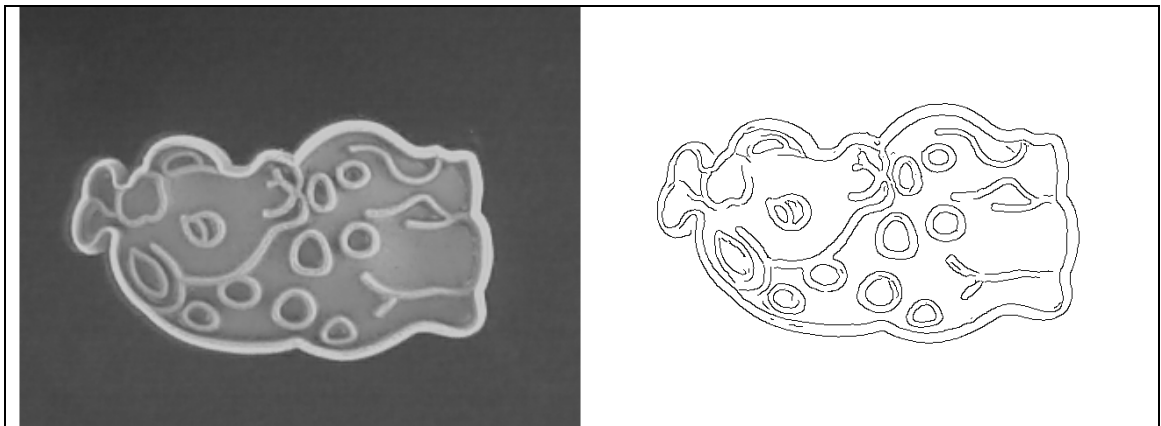
<sup>2</sup> See Section 4.1

## 7.7 'Real World' Objects

The objects used as examples so far have presented simple geometric outlines used to demonstrate the capabilities and limitations of the proposed methods. In this section two objects are presented that are similar to the sort of object that may be scanned during commercial use.

### 7.7.1 Giraffe Cookie Cutter

The giraffe cookie cutter mould is a complicated object consisting of many curved sharp edge of varying degrees of curvature. The edges are 'sharp' near-vertical surfaces. The exterior (6mm) edges are slightly higher than the 'pattern' edges (approximately 4mm), however the holes in the pattern edges are relatively small and many of them present difficulty for the scanner to approach at any orientation without causing occlusion due to other nearby edges. In addition, the close proximity of many edges at conflicting orientations means that this object is a real test of the ability of the partial scan methods to resolve the problem of selecting the correct scan orientation.



**Figure 7-36: Giraffe Cookie Cutter Mould (left image) presents a challenging task for the determination of optimal scan orientation. Right image shows detected edges using the Canny operator (Gaussian  $\sigma$  1.1  $T_{\min}$  29%  $T_{\max}$  89%).**

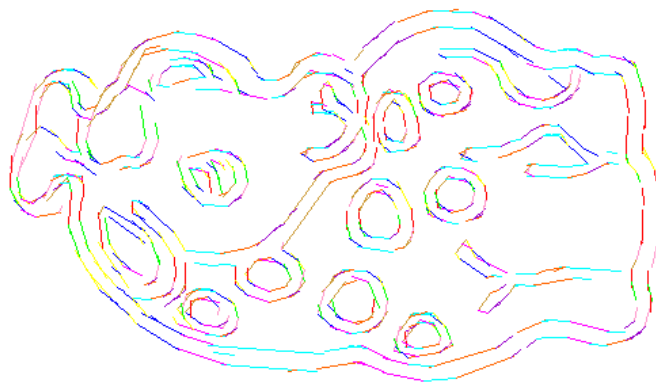
The object provides a good test for the edge detection because it requires the detection of internal edges within the outline of the shape (rather than just the detection of the perimeter). Providing a clean edge detection of the giraffe image proved difficult as spraying the object caused a loss of contrast between the internal edges and the base. Lighting the object without shadows being cast (that would create false edges) at misleading orientations within

the critical image area also proved difficult. Another problem was caused by the camera, which displayed a degree of parallax error across the object, so that it was possible to see the external sides of the cutter edges; thus complicating the edge detection (see Section 4.3.2.1).

The quality of the edge detection was not quite as good as was hoped for, with some loss of edge information around the internal edges due to lack of contrast, but the parameters used in the example shown in Figure 7-36 provided an acceptable determination of the majority of edge positions.

### **Cookie Cutter Vectorisation**

Vectorisation of the cookie cutter required a tight correspondence between the detected edges and the vectors around quite 'tight' curves. This was achieved by setting the minimum permitted edge to a short length of 5 pixels with a corresponding restricted angle tolerance of  $9^\circ$ . In order to prevent vectors 'bridging' the short gaps between different edges the bridge gap limit was also set to a short length (5 pixels) although this meant there was a good chance that any 'accidental' gaps caused by the vectorisation process were unlikely to be joined. The result of this combination of parameters is shown in Figure 7-37 and was achieved in 61.41 seconds. This is significantly longer than for previous tests due to the large number of short vectors that must be considered



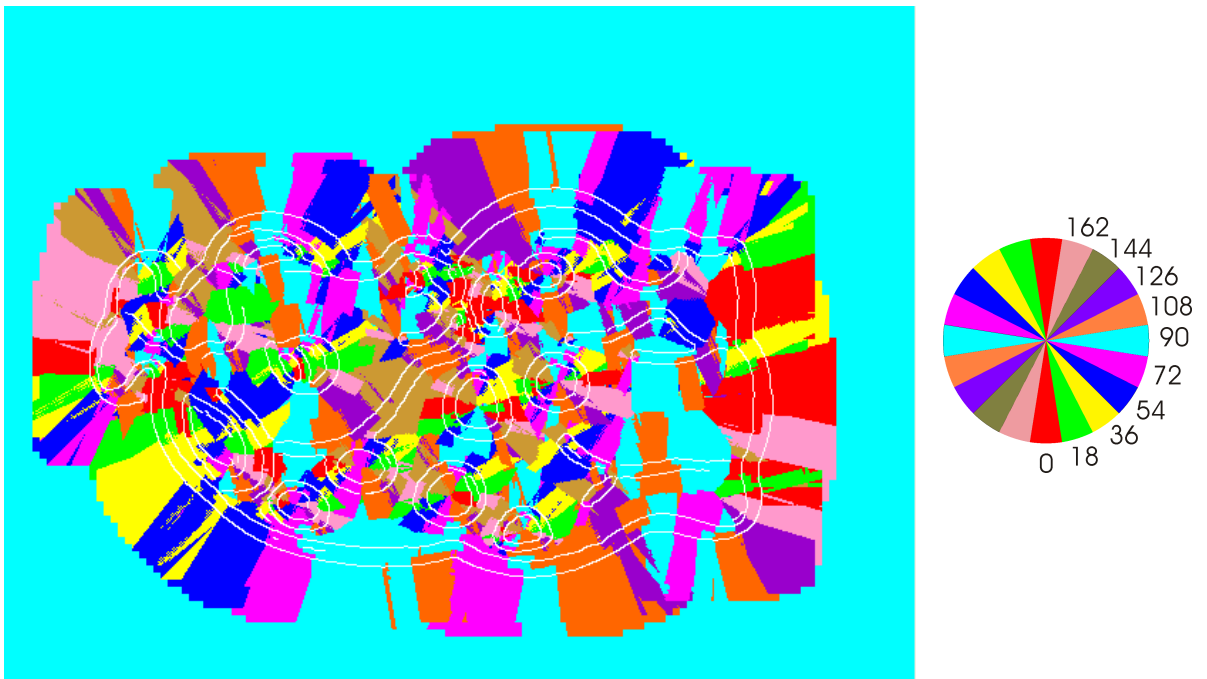
**Figure 7-37: Least Squares Vectorisation of giraffe object**



### Cookie Cutter Scan Region Development

From previous experiments it was obvious that the Scan Region Orientation method produced poor results when regions of different orientations are overlapping. The results of the SRO method are therefore not shown here.

The nearest vector orientation map does not have the problem of overlapping scan areas. The scan region map for the NVO method (Figure 7-38) shows the complexity of many small scan regions caused by the close proximity of the vectors around the circular patterns. The larger outer regions track quite well around the perimeter of the mould, although some odd, narrow scan regions have been created by short vectors around the perimeter that could not be removed by the vector-merging process due to the necessity of keeping short vectors to preserve the detail of the inner regions; these are not a significant problem as they are of an orientation similar to the surrounding region, however in the 'real' system these would add a significant time to the overall scan as each would have to be treated as a separate region.



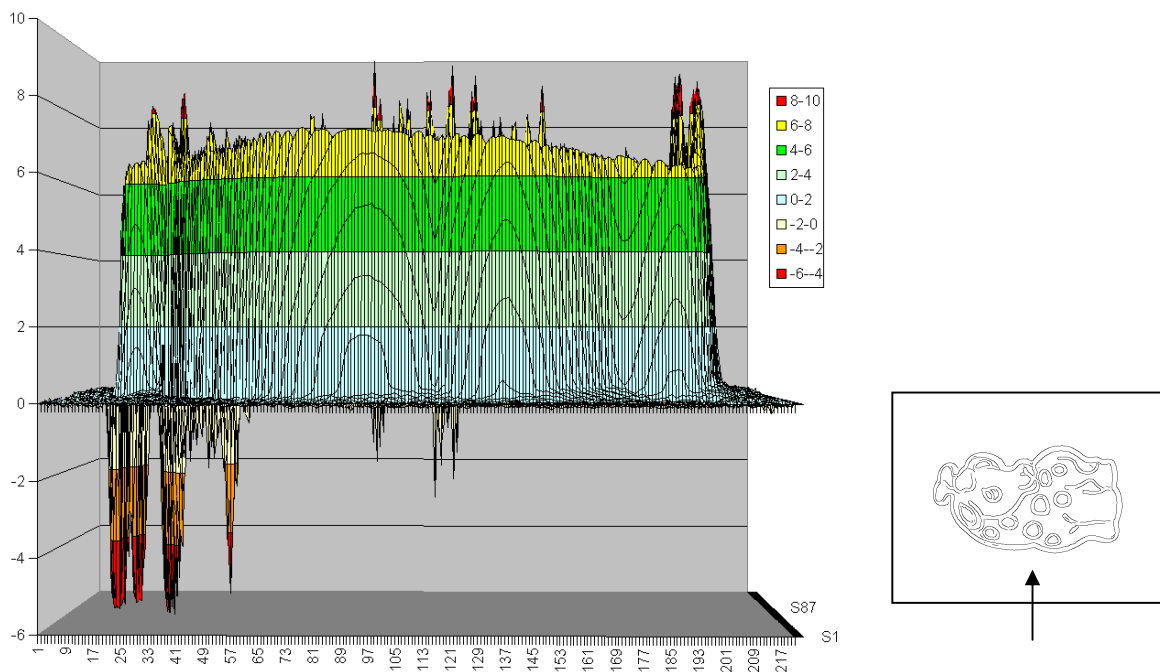
**Figure 7-38: Output map by nearest vector orientation with edge detection overlay. Colour wheel indicating regions of edge orientation shown to right.**

The dominant scan direction is the 90° scan, which is as expected the orientation most aligned with the longer axis of the giraffe mould. As shown in Figure 7-39 and 7-40 this scan orientation shows strong occlusion errors along the edges perpendicular to the primary scan direction. These errors are especially strong at the exterior edges around the 'head' and 'feet' of the giraffe, but also exist around the interior circular edge patterns. Bow wave effects are minimal around the object in comparison to the magnitude of the occlusion errors.

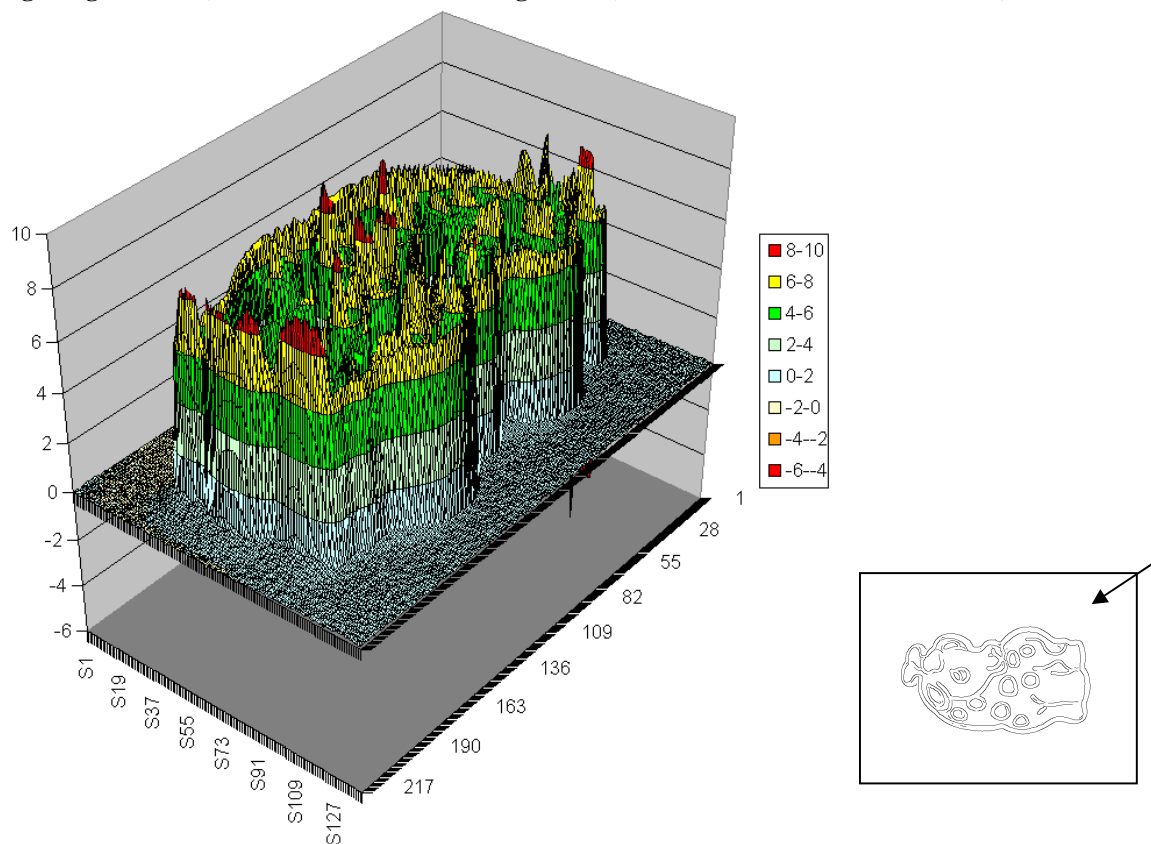
The results of the scan by the Nearest Vector Orientation method show much improvement over most of the scan area when compared to the best single orientation method; however some obvious errors remain in the scan. The major errors at either end of the object have been greatly reduced. Other spikes are also associated with sudden changes in orientation around the perimeter and the internal circular regions although there are also places where the combined partial scan has 'imported' incorrect data.

These errors appear to be worse as the distance from the scan calibration origin increases, and are therefore likely to be due to poor calibration of the scan with respect to the image map.

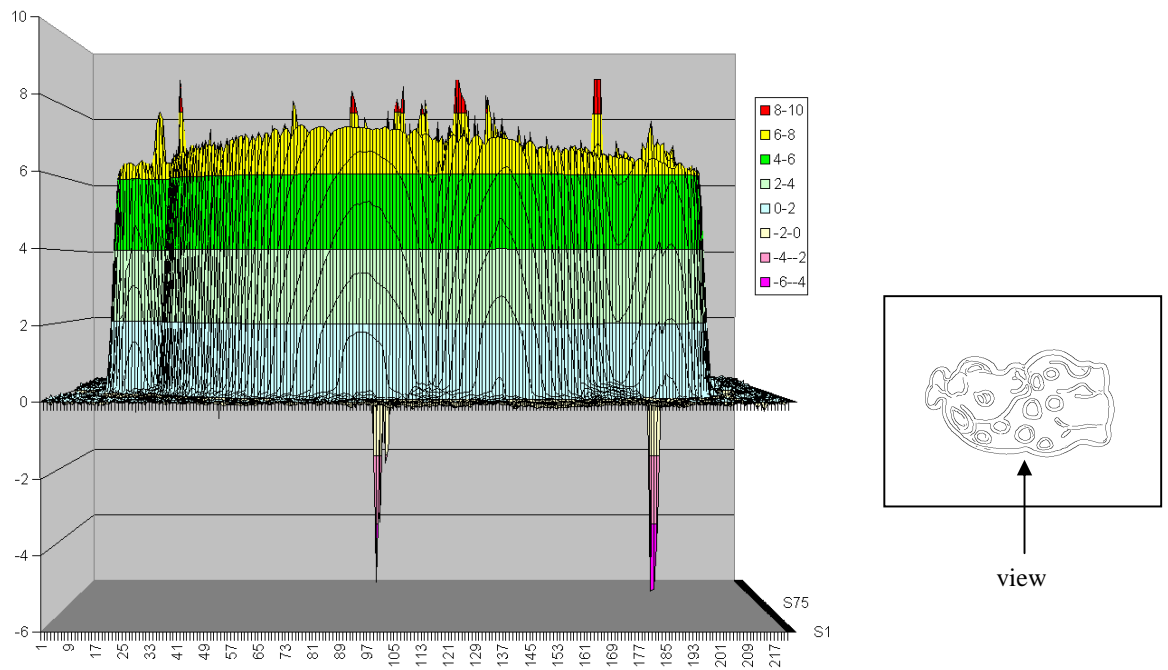
Because many of the scan areas are small their alignment with the scan image is critical, as an error of only a few millimetres on an object such as this leads to the accidental selection of an orientation that is inappropriate at the intended location. Suppression of bow waves is good across the image as the full region width can be applied without concern for overlapping regions.



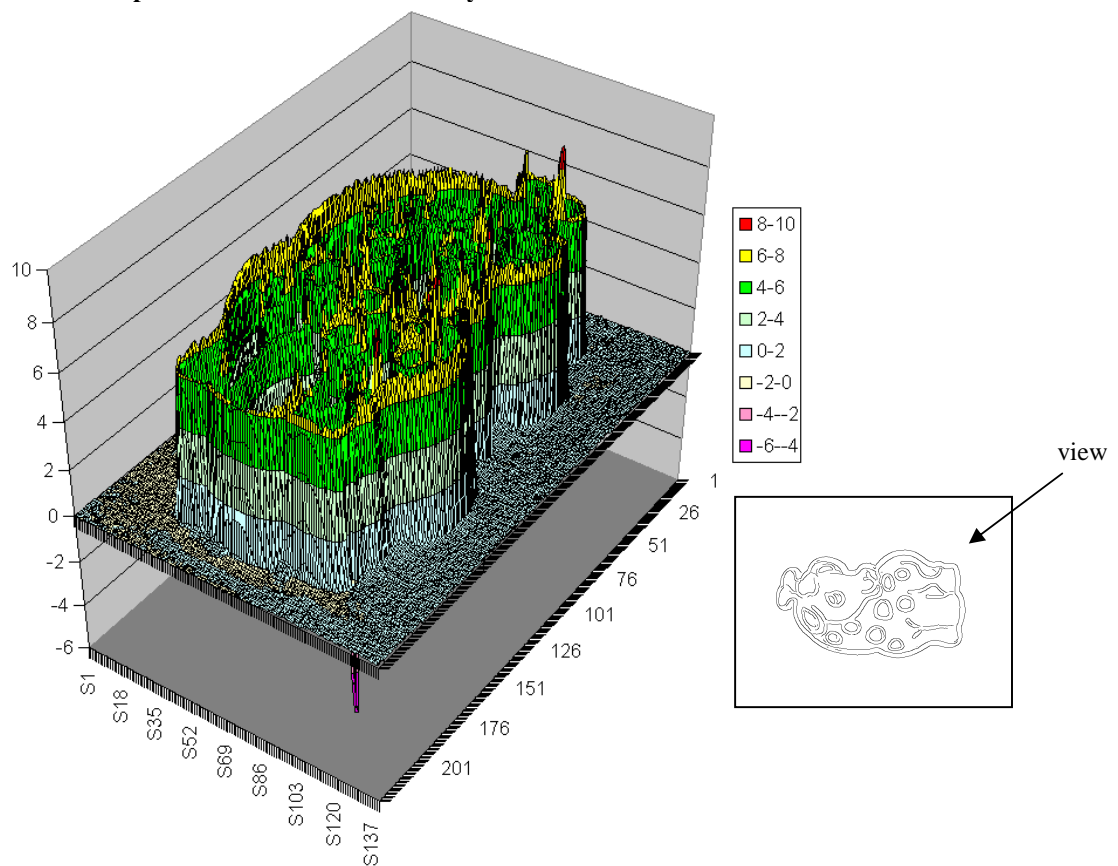
**Figure 7-39** Side view (head to left) of 'best' scan orientation shows magnitude of spikes (actual object edge height is 6mm) Bow wave also visible at right side (see inset for relative view direction)



**Figure 7-40** Oblique view from 'top right' corner of scan image looking down on scan shows many large errors in the 'best' single orientation scan. (See inset for relative view direction)



**Figure 7-41: Side views shows reduction in bow waves but also the magnitude of the spikes which have been imported to the combined scan by selection of incorrect orientation**



**Figure 7-42: Oblique view of scan selection by nearest vector method shows reduced bow wave around area of 'feet' (RHS of image).**

### **Cookie Cutter Results for Nearest Vector Orientation Method**

Figure 7-41 and 7-42 show views of the combined output for the Nearest Vector Orientation method. For reference, the origin of the scan calibration is located at the bottom left corner of the inset images that indicate the view direction of the scan

It proved difficult to determine the correct orientation at many positions within the cookie cutter object due to the complexity of the edge map. Poor calibration caused by webcam distortion issues, especially parallax error was (at least in part) the cause of the resulting errors. This had a particularly serious effect on the results of the Nearest Vector Orientation method, as an incorrect orientation selection causes incorrect data to be selected that is unmitigated by the averaging process used by the Scan Region Orientation method.

The detection of very small regions on this object would cause a large number of extra regions to be scanned in a system where the output of the software was a scan control path. Many of these regions are very similar in orientation to the regions that surround them and could perhaps in many cases be subsumed by that region without loss of detail.

### 7.7.2 Bronte Plaque

The plaque is a highly complex and challenging object that has edge information at a number of different scales. There are major geometric edges associated with the general elliptical shape of the plaque and minor geometric edges associated with the detail of the plaque, specifically the raised profile of the house, gravestones and the 'scrolls' on which the text is printed. The window panes of the house are also very small geometric edges. There are also many 'textural' features such as the stippled effect representing the trees, grass and paving. Finally there are edges that are 'transitional' edges, indicating a change in reflectivity of the surface (i.e. the text printed on the scroll area).

The major axis of the elliptical plaque is 105mm and the minor axis is 75mm. The circumference of the plaque is of height between 12 and 13mm (there is some variation around the plaque). The edge is also slightly 'undercut' in profile so the top of the edge overhangs the bottom. This 'negative' edge angle is likely to have some effect on the magnitude of secondary reflection errors around the object. The scroll at the top and bottom extend above the edge height at either end of the scroll and curve downwards towards the centre.

The object was *not* sprayed with the Flawfinder Developer spray as this would prove very difficult to remove from the texture detail and also disguise the effect of the transitional edges.

In this test scan the scan point resolution used was only 0.5mm (i.e. 2 points per mm in x and y dimensions), which is not really sufficient for an object with surface information at such a small scale; however the errors that the technique aims to remove are apparent even at this coarse scan resolution. At this low scan resolution this object required 21.5 minutes to capture the data area. Scanning at a higher resolution significantly increases the total scan time.

The poor quality optics of the webcam combined with the size of the object resulted in an image with poor focus causing a loss of detail over a large part of the image, so the digital camera image was used for this object. Figure 7-43 shows the image used for edge detection of the plaque.

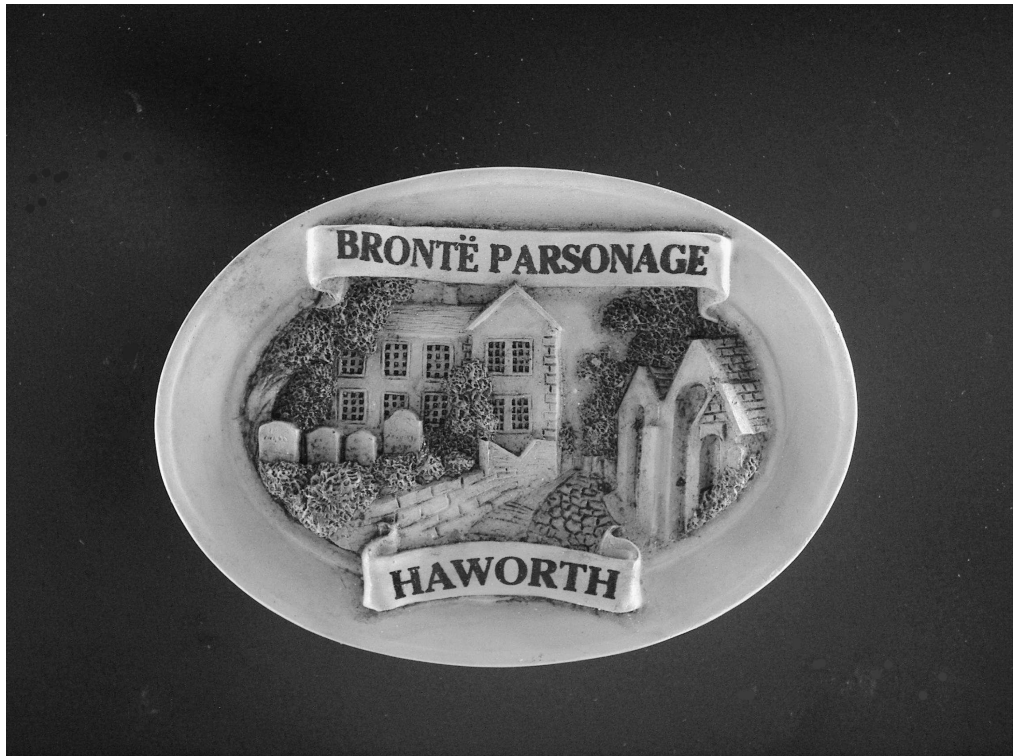


Figure 7-43a) Brontë plaque 640x480 greyscale image from digital camera used for image processing.



Figure 7-43b) Plaque shown from side to illustrate undercut of edges and profile of surface detail

### Brontë Plaque - Edge Detection

Edge detection of the plaque was performed using the Canny edge detector with the following parameters: Gaussian smoothing  $\sigma$  1.1,  $T_{\min}$  28% and  $T_{\max}$  88% as shown in Figure 7-44a). The output shows good correspondence with the majority of geometric edges in the plaque, although much of the textural information is also detected. There are also some edges caused by highlights and shadows cast by the features of the object. Using a higher Gaussian smoothing parameter removes more of the edges at a textural scale but also causes some loss of more significant detail. Figure 7-44b) shows the results with the Gaussian smoothing changed to 1.8. This also causes some slight displacement of the edges. Given the level of detail of the object required by the final scan there appears to be no single set of parameters that can capture just the required information at the different scales.

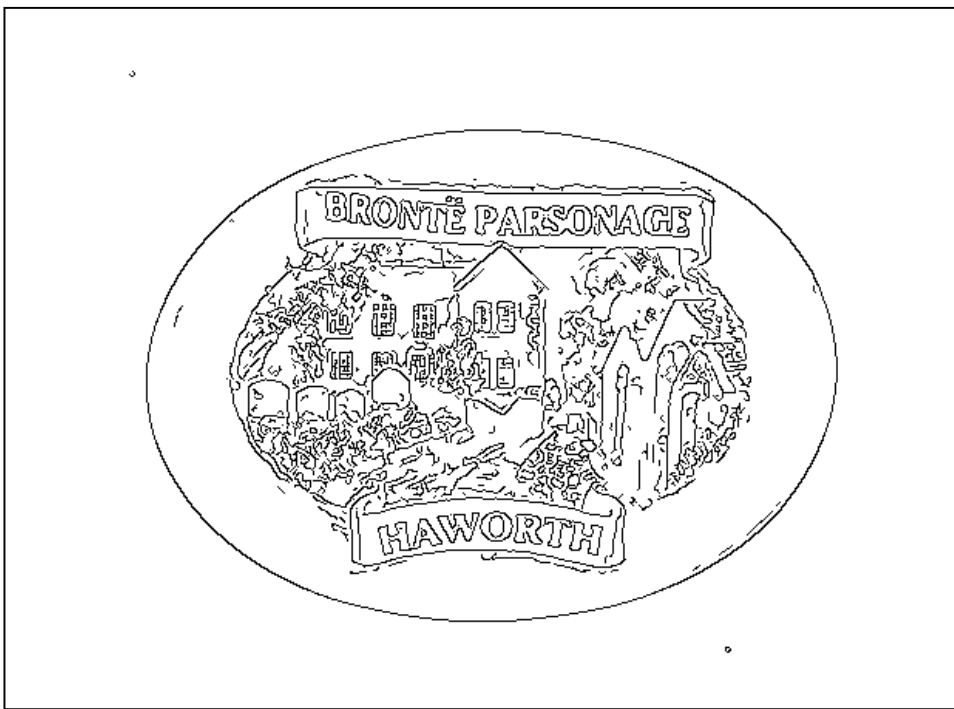


Figure 7-44a) Output from Canny Edge Detection using parameters  $G\sigma 1.1$ ,  $T_{\min} 28$ ,  $T_{\max} 88$



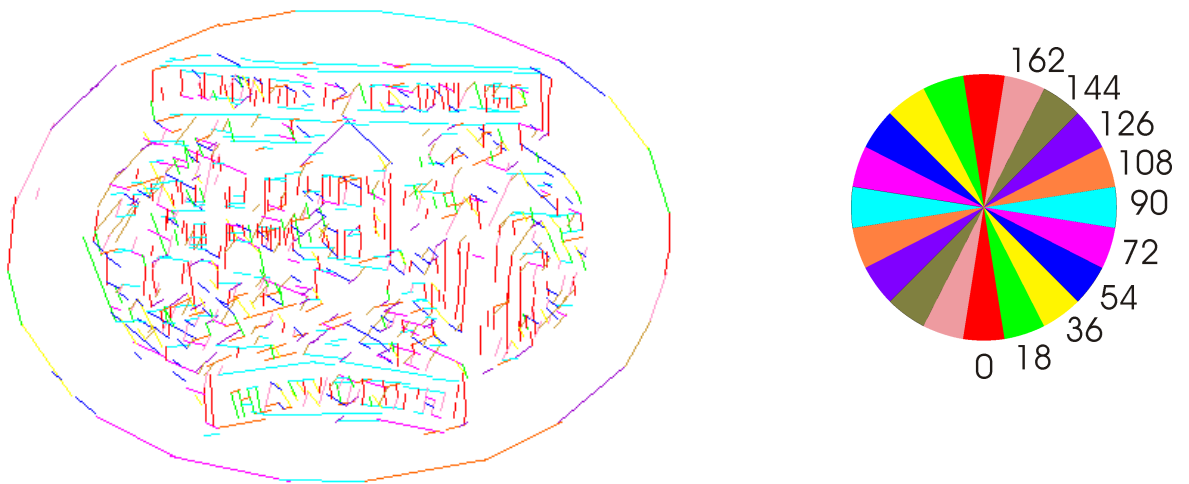


**Figure 7-44b) Output from Canny Edge Detection using parameters  $G\sigma 1.8$ ,  $T_{min}28$ ,  $T_{max}88$**

### **Brontë Plaque - Vectorisation**

The vectorisation process requires parameters that are quite strict in order to maintain a tight correspondence with the detected edges. Separating the ‘true’ edge vectors from the textural edges is difficult without losing also details that need to be preserved.

Figure 7-45a) shows the large number of potential vectors that may be created from the edge detection using the image in Figure 7-44a) as a base for vectorisation. This results in the process taking over 2 minutes. By comparison, the corresponding vectorisation of the image shown in Figure 7-44b) takes only 63 seconds to compute. However this vector map shows a less accurate correspondence between the edges and the vectors (as shown in Figure 7-45b).



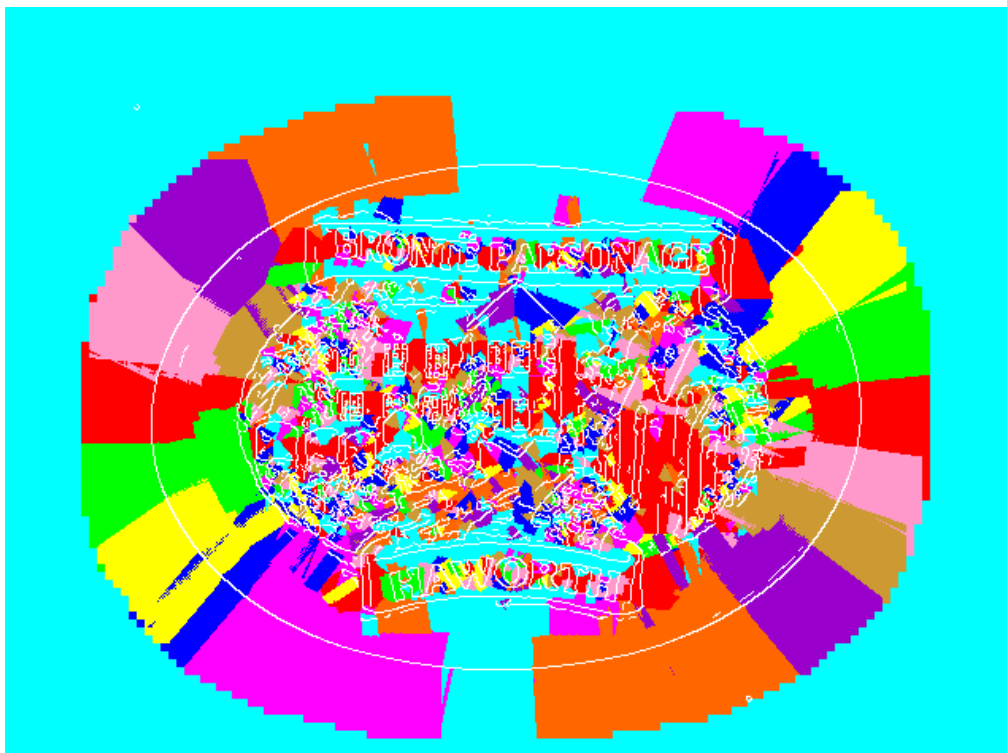
**Figure 7-45a): LS Vectorisation 5-10-5-5 – vectors shown in 18 degree bands from 0-180° (see key) requires 2min 18 seconds to complete.**



**Figure 7-45b): LS Vectorisation 5-10-5-5 using edge detection image with a greater smoothing takes less time to compute due as there are fewer edges, however the resulting vectors show less correspondence to the edges discovered by the Canny detector.**

### **Brontë Plaque – Output Map by Nearest Vector Orientation**

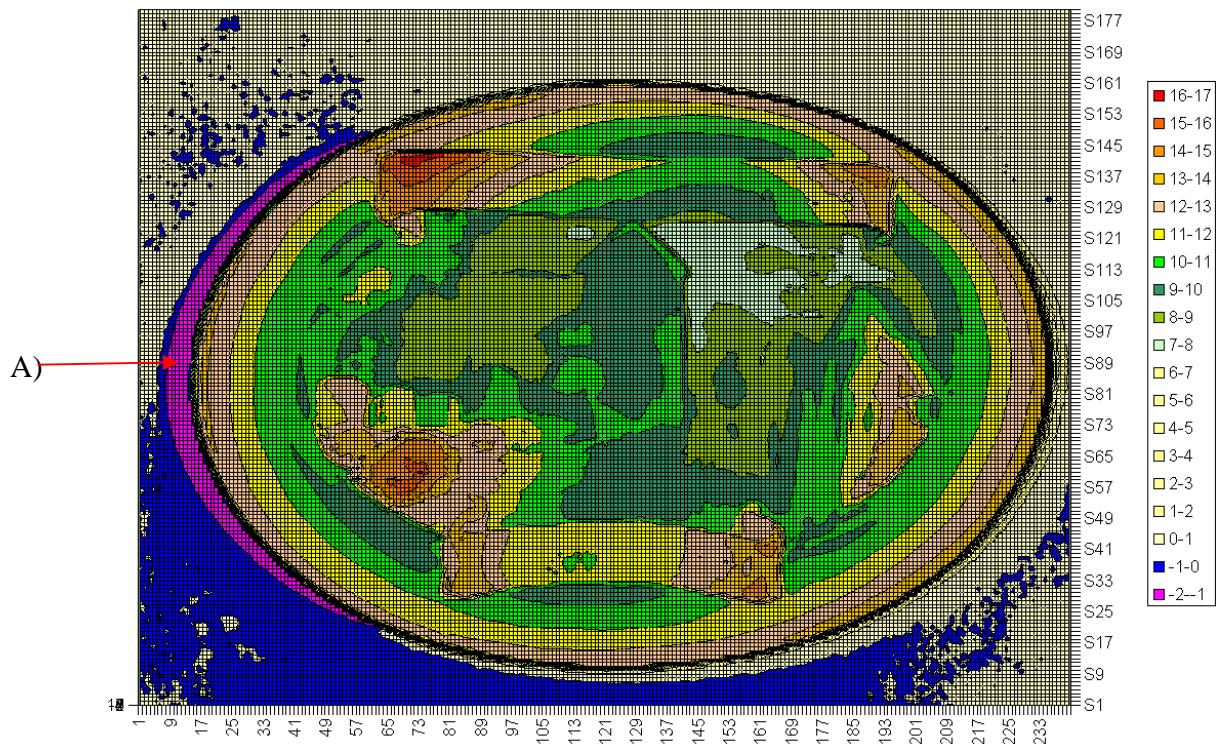
The vector map shown in 7-45a) is used to develop the nearest vector orientation scan region map shown in Figure 7-46. The output map shows good correspondence between the orientation regions and the original detected edges. The orientations of 'large scale' edges are well-indicated however the detail of the object is somewhat confused due to the large numbers of short vectors. Furthermore, the processing time required by this method is very high (7 minutes, 13 seconds) due to the large numbers of vectors in close proximity that must be considered at each point. Many of these scan regions may be so small as to be ineffective in selecting the correct orientation and might possibly be ignored without impact on the quality of the scan. Although the time taken calculation of the region map is considerably longer than for previous objects, a single scan of the plaque takes 21.5 minutes, even at a coarse resolution, and improvements in computing power and refinements and optimisation of the code would further reduce the time to calculate the map.



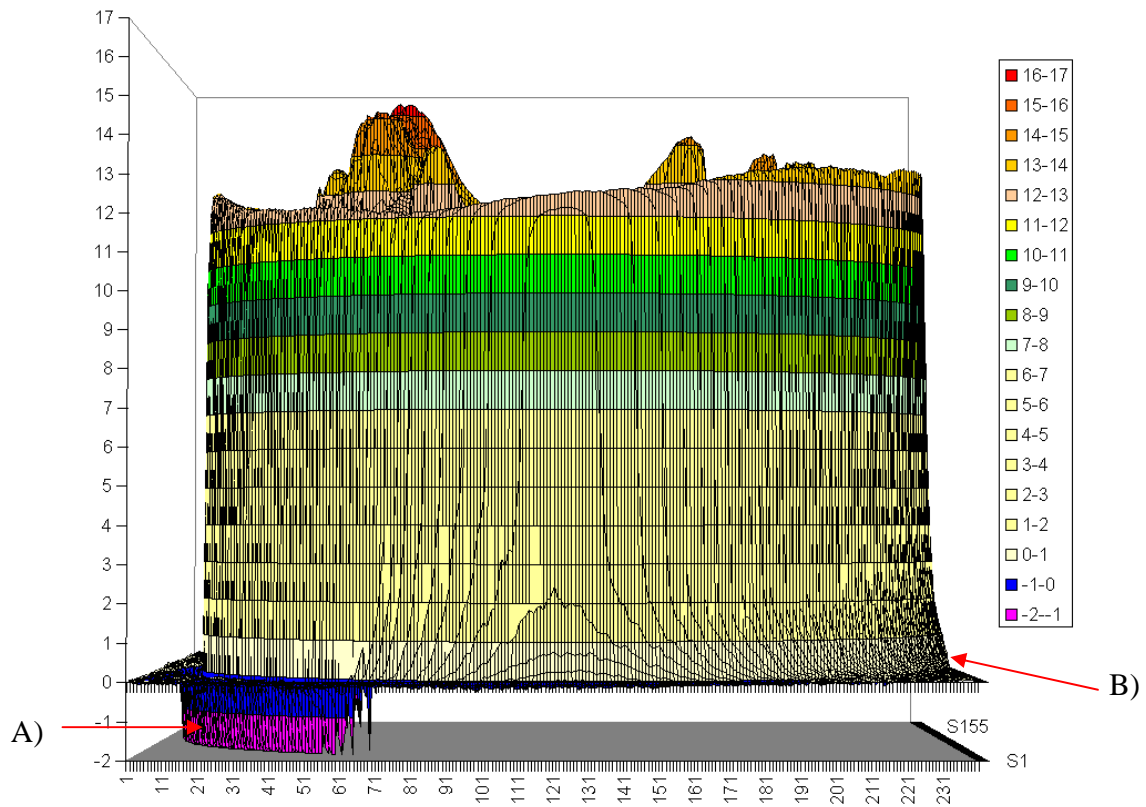
**Figure 7-46: Output map by Nearest Vector Orientation shows a good orientation correspondence at the cost of long processing time**

### Brontë Plaque Scan Output - Best Single Scan Orientation

The indicated best single scan orientation for both methods is the 90° scan (i.e. with the sensor parallel to the x-axis) which is as expected. Figures 7-47 and 7-48 show two views of the single best orientation scan. The output from this scan shows trough spike output around much of the curve of the perimeter of the object to the left side of the scan (error A in figure 7-47 & 48). The ‘undercut’ of the outer edge of the plaque has helped to reduce the amount of secondary reflection, however it has resulted in a little ‘buttressing’ of the side of the object (error B in figure 7-47 & 48) and some bow wave appears at the edge of the scan area. As the laser cannot produce the true profile of the edge from the current scanning angle it is a moot point as to whether the external edge profile is of much importance, however the ‘best’ result that could be hoped for would be a ‘vertical’ edge. It is difficult to see whether there are any errors in the internal surface of the scan as there is no ‘ground truth’ scan to which the scan profile can be compared.



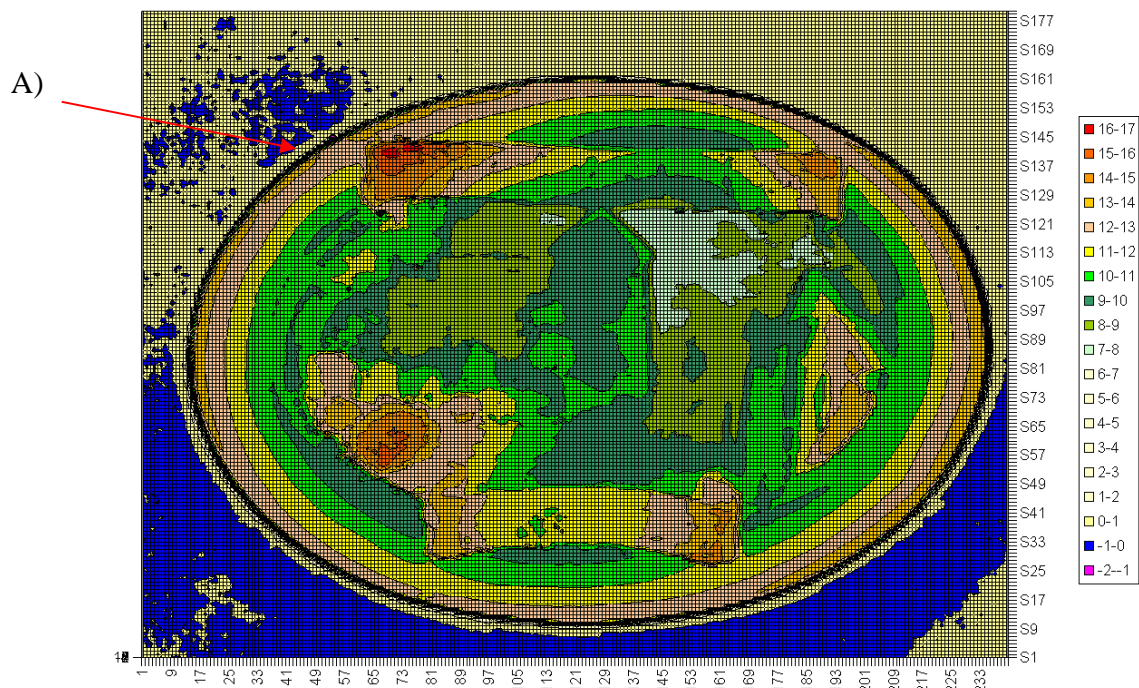
**Figure 7-47: Top-down view of best single orientation scan (90 degrees), scanner orientation parallel to x-axis. magenta region A) corresponds to the negative trough**



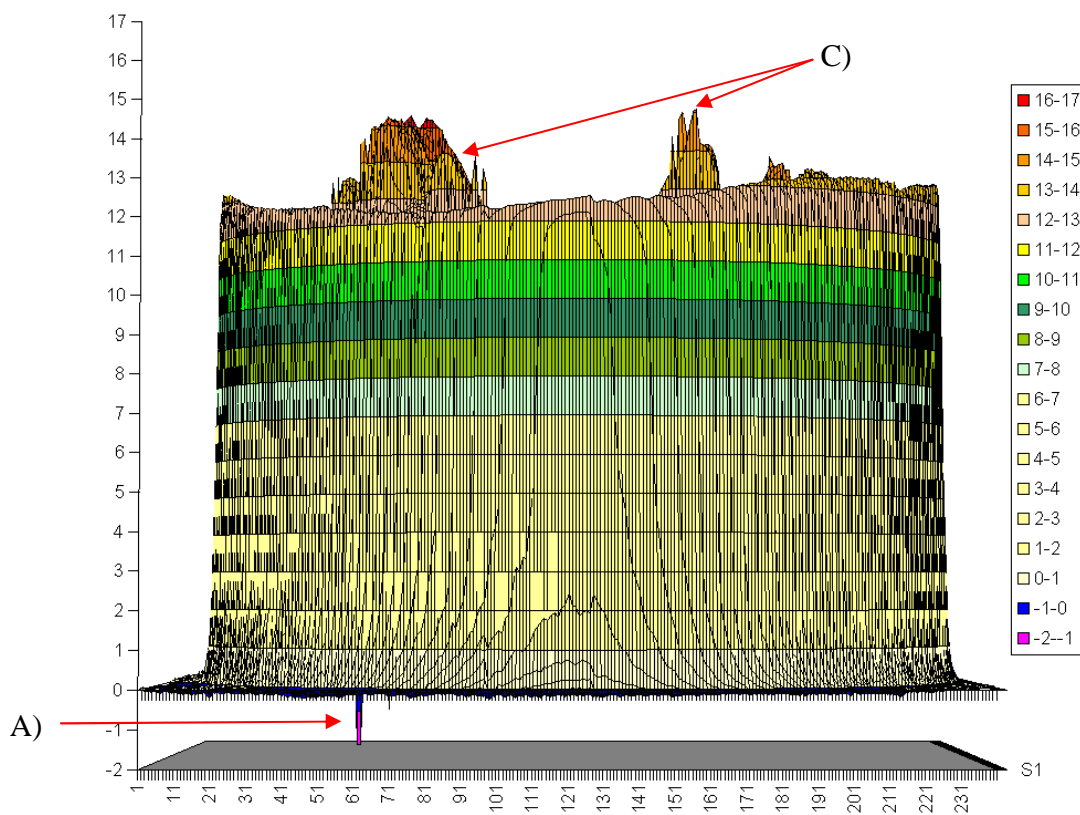
**Figure 7-48: Front edge view of best single orientation scan (90 degrees), scanner parallel to x axis. Negative trough spikes can be seen at left side of scan (A). At the right side a slight outward distortion of the edge can be seen where the secondary reflection has caused some thickening of the side (B).**

### Brontë Plaque - Scan Output by Nearest Vector Orientation

Output from the Nearest Vector Orientation method is shown in Figure 7-49 to 7-54. The result of the NVO method shows good removal of the major geometric scan errors around the edge of the object, with only one small negative spike noticeable (where it has been left behind due to a tiny region of the main orientation scan being used). The buttressing effect around the right side of the object has been entirely removed and (unlike the SRO method) no other errors have been introduced around the edge. The surface detail of the plaque is difficult to analyse due to the small scale of many of the scan orientation regions, however it is possible to see where some data spikes have been imported from scan orientations around the area of the scroll edges. This may be due to slight calibration misalignment or confusion between the regions developed based on true 'geometric' edges and those based on transitional edges created where edges were detected based on shadows cast by the true edge.

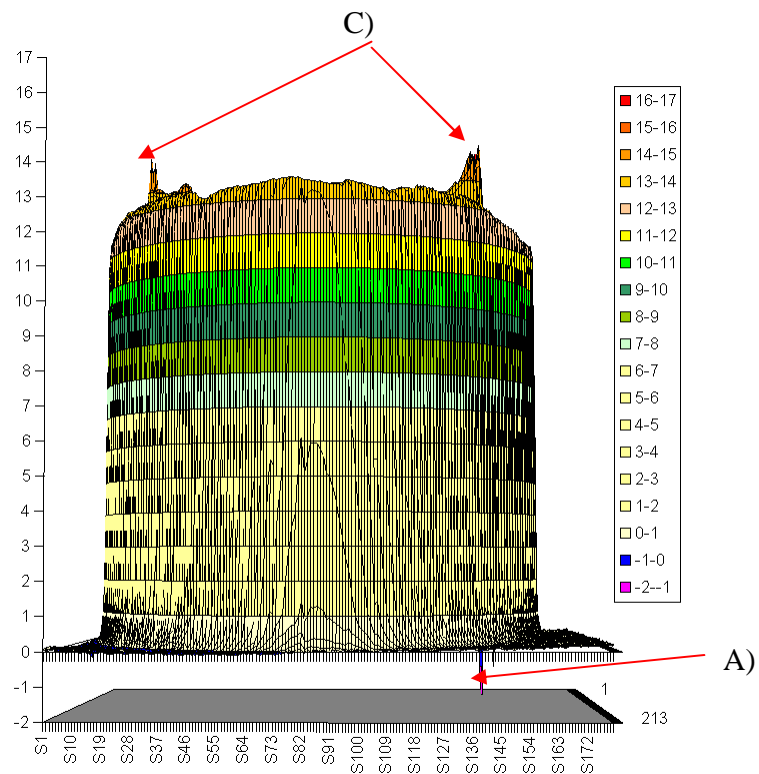


**Figure 7-49: Overhead view of combined output by nearest vector orientation shows reduction in serious trough errors around left side compared to the 90 degree scan., Troughs are almost completely eliminated except for a small spike at position (A).**

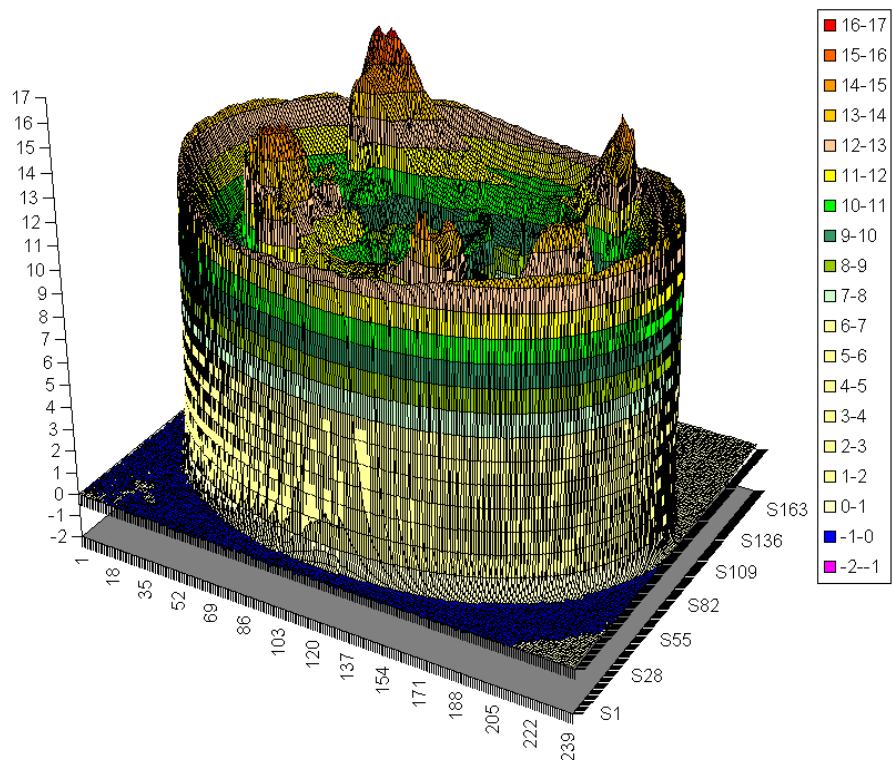


**Figure 7-50: Front view shows small negative spike error to rear of scan (A). Note that more 'spikes' are associated with edges on top surface of object. (C)**



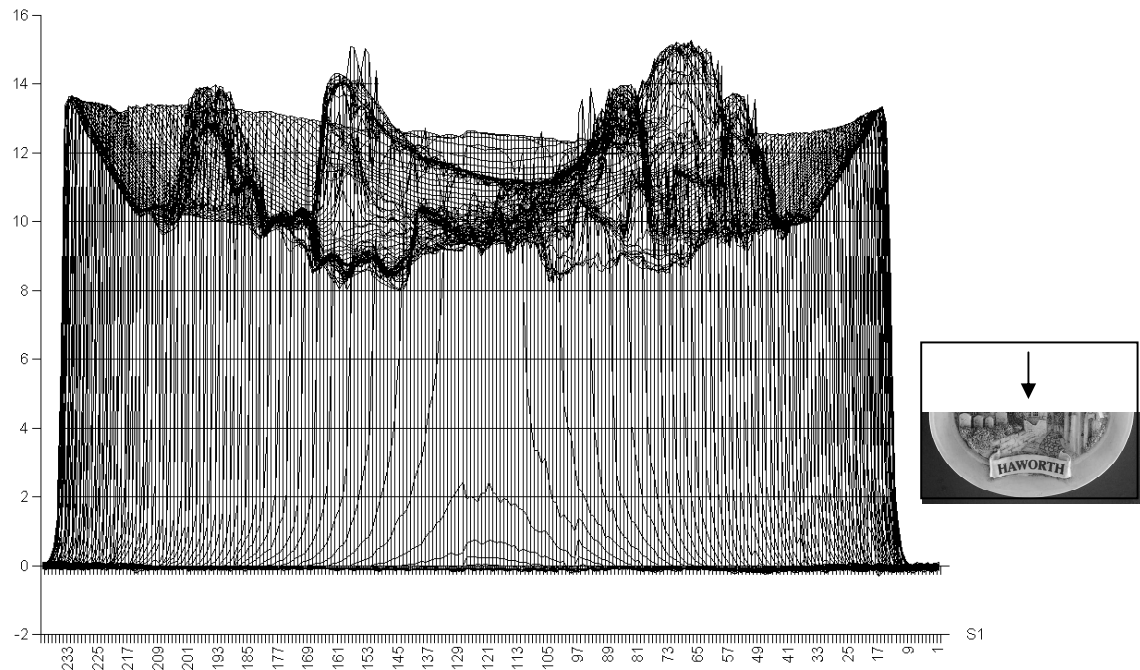


**Figure 7-51: End view of plaque shows spikes on bottom edge (A) and top surface at edges of scroll (C). Bow wave and buttressing problems removed**

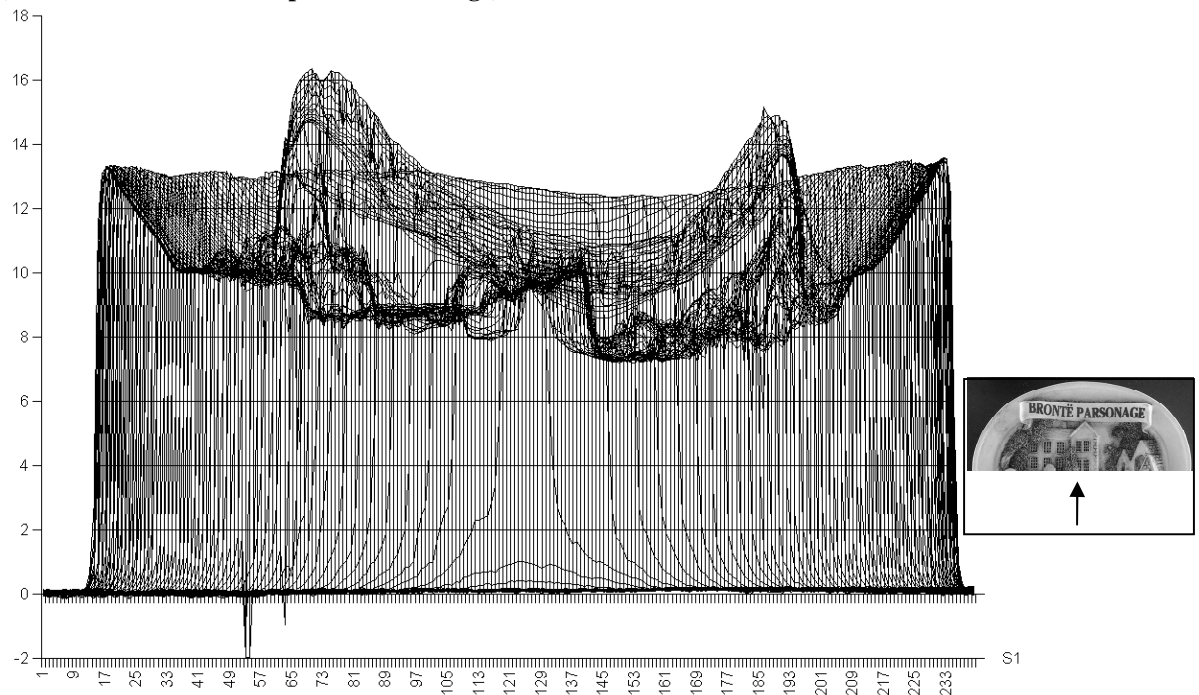


**Figure 7-52: Isometric view of plaque combined scan by nearest vector orientation shows some spikes on top surface.**

Figure 7-53 and 7-54 represent the cross-sectional view along the x axis at a midpoint on the scan. This allows some detail of the central region of the scan to be seen in profile without the raised outer edge obscuring the view. In each case the inset represents the direction of view.



**Figure 7-53: 'Front' half-slice (corresponding to bottom half of image) cut viewed from back (inset section view from top of camera image)**



**Figure 7-54: 'Back' half-slice (corresponding to top half of image) cut viewed from front (inset section view from bottom of camera image)**



## Brontë Plaque Summary

The output from the Nearest Vector Orientation method removes the majority of the geometric scan errors. Although the method produces a clean result overall, it does show local distortion where errors are missed due to poor alignment of the scan regions. This may be due to calibration or an artefact of the vectorisation process where the vectors have grown too far. The SRO method disguises the magnitude of such errors where data in the overlapping scan regions are averaged; averaging the overlapping data regions also reduces the quality of detail in those areas. Although the NVO method takes far longer, due to the large number of edges that must be considered, the total time required to produce the scan map is still much less than the time required for scanning the object at even a course scan resolution of 0.5mm per sample. When the correct data is selected the accuracy is as good as the 'best' single scan, however calibration is critical with so many small orientation regions. Further work is required to determine 'true' geometric edges from edges caused by 'noise' (i.e. texture).

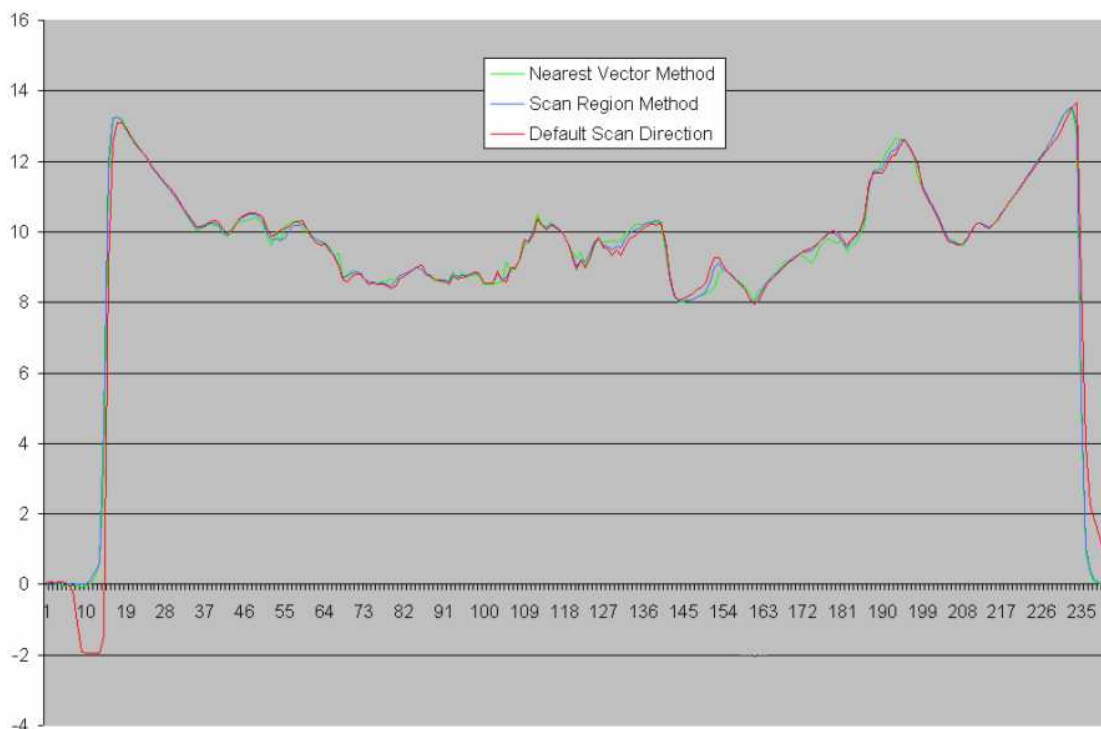


Figure 7-55 shows a comparison of different scan methods on single cross-sectional scan line through the centreline of the scan on the x-axis. Both combination methods show a marked improvement in removing the major trough and buttressing effects compared to the default scan.

## **7-8 Summary of Results**

A highly contrasting image allows simple determination of image thresholding but the low surface reflectivity of the object can seriously affect the quality of the scan data. It may be necessary to spray the object but this can disguise small details and make thresholding of the object more difficult. Objects with a highly specular surface require spraying to reduce secondary reflections.

Careful calibration is required for camera alignment and compensation for optical distortions in order to adjust the camera image. Without such adjustments the scan region map, generated from the image, cannot be successfully related to the scan space and so the laser may be incorrectly oriented at critical locations (such as the junctions where different orientations meet). The calibration methods applied in the test software were not fit for purpose as they did not account for radial distortions. This is a limitation of the optical hardware tested and not (in itself) a fault of the 'region orientation' methods under investigation. Corrections have been successfully applied for radial lens distortions in many other applications [122,123,124].

It has been shown that, when the data is correctly calibrated, the combined scan methods produce composite scan output with a lower average error than any single orientation scan (See Figure 7-19), with the NVO method showing the lowest mean square error.

The advantage of the Scan Region Orientation method is that the method is fast to generate a selected 'best' orientation. However the SRO method often shows poor results where scan regions of greatly different orientations overlap. Results from straight edges and those of relatively low curvature are generally acceptable but the required minimum region width also often causes overlaps between disjointed orientations where sudden changes in edge orientation are present and overlaps at geometric corners of an object always present a problem.

The Nearest Vector Method is slower to generate output, although it is still able to determine a result much quicker than for a physical scan of the object. The results are generally better than the SRO method and are unlikely to be worse. Generally the NVO method is more likely to cause localised distortions. Where SRO averaging is used it tends to disguise the distortion or extend the problem over a wider area (due to the required minimum region

width). Any errors remaining after the NVO method has been applied are easier to find and could be removed with a 'spike removal / hole-filling' post-process step.

To achieve good results more quickly it would be possible to apply a combination of the SRO and NVO methods whereby the SRO method was applied initially and then the NVO method applied only in the areas where regions of (significantly) different orientations overlap. The time-savings made by limiting the use of the NVO method to areas of overlap would obviously depend on the percentage of the image with overlapping regions and the number of vectors involved.

The NVO method also requires accurate camera calibration, as very small calibration errors can lead to the selection of an incorrect orientation at any given point. This is especially critical in locations at geometric corners, junctions or vector intersections where the orientation changes significantly. At such positions it may also be advantageous to apply Wong's original method to a very small area around the junction. This is discussed further in Chapter 8.

Problems occur with regions of high curvature (such as internal holes) that are not necessarily related to the method itself but to limitations of the triangulation laser when the return signal is occluded at all orientations of the sensor. The user could be alerted to the occurrence of such situations so that other methods could be used.

---

<sup>122</sup> R. Cucchiara, C. Grana, A. Prati, R. Vezzani, "A Hough Transform-Based Method for Radial Lens Distortion Correction," ICIAP, p. 182, 12th International Conference on Image Analysis and Processing (ICIAP'03), 2003.

<sup>123</sup> Satoru Yoneyama, Hisao Kikuta, Akikazu Kitagawa, and Koji Kitamura, "Lens Distortion Correction for Digital Image Correlation by Measuring Rigid Body Displacement," Opt. Eng. Vol. 45, Issue 2, Feb. 2006

<sup>124</sup> Toru Tamaki, Tsuyoshi Yamamura, Noboru Ohnishi, "Unified Approach to Image Distortion," *ICPR*, p. 20584, 16th International Conference on Pattern Recognition (ICPR'02) - Volume 2, 2002.

## Chapter 8 Future Work

This chapter contains ideas that may be investigated to improve the quality of the results, as well as some factors which have become apparent through testing that would be mandatory requirements for a full working system.

The calibration methods used in this project were insufficient for the purpose. The affine transform proved to be a poor model which, with the benefit of hindsight, should have been realised at an earlier stage. Corrections for radial and perspective distortions in the image must be applied prior to the registration and image analysis processes.

### 8.1 Image Analysis

Further development of the edge detection methods to provide a ‘cleaner’ edge image that is less dependent on the selection of correct user input parameters. The edge detection algorithms employed were based on greyscale image analysis which has been the standard in image detection for many years. Despite the common use of colour film and digital imaging technology, most of the approaches to colour edge detection attempt to extend a greyscale edge detector to colour images [125]. However there are some methods that take a different approach such as Ruzon and Tomasi’s [126,127,128] Generalised Compass Operator and work on Photometric Quasi-Invariants (van de Weijer *et al*)[129].

Corners in the image scene represent locations where selection of the correct edge orientation proves difficult because there are potentially two or more competing orientations. Without further work it may be impossible to determine the best orientation for a point with the region of influence of a corner or junction. However, if these regions can be flagged as potential areas of difficulty prior to the scan, then it may be possible to resolve these small localised regions by reverting to a previous method. Wong’s method was shown to be effective at removing occlusion and reflection errors but requires a full set of orientation scans in order to identify the ‘good’ and ‘bad’ scan data. Whilst this proved prohibitive in terms of total scan time, it may prove useful on a localised basis if the areas that require scanning in all orientations are small with respect to the total scan area.

A number of corner and junction detection techniques have been proposed with varying degrees of success. Although simple corners are relatively easy to detect given a ‘clean’

image, it is difficult to define exactly what a ‘junction’ is in an image – there are many possible arrangements of intersections that must be considered.

Shadows in the image often cause problems in line following and segmentation of real world images. Although shadow removal is a recognised and real problem we can seek to avoid it being a major factor by controlled lighting of the scene. In practice however the complete eradication of shadows in the scene proved very difficult.

Other approaches, such as region detection and texture analysis, might also be investigated as alternative or auxiliary methods to edge detection as a means of determining the object position and geometry. The potential for using a combination of edge, region and texture detection methods, along with some means of reconciling the different edge models, to provide a consensus output model may prove successful.

## **8.2 Improved Vectorisation Methods**

The vectorisation process described in Chapter 3 and explained more fully in Chapter 5 provides the means by which the edge orientations are identified to the laser scanner. Whilst this process works to a reasonable level in most cases, there is certainly room for some improvements.

The Least Mean Squares vectorisation method used in this project requires that each edge pixel in the image (that has not already been included in a vector) may be considered as the initial point of a vector and as such an attempt will be made to ‘grow’ that pixel into a vector. In many cases this test vector will not meet the minimum requirement parameter for vector length and will be rejected. This ‘bottom-up’ approach is time-consuming in the early stages but provides good localisation to the position of the edges in the image.

In contrast, the vectors provided by the linear Hough Transform method do not contain any knowledge of where the vectors begin and end within the image space. Attempts were made to localise the vectors by providing a ‘moving window’ on the image and combining the resulting vector segments was only partially successful in localising the vectors because there was a minimum effective window dimension before the windowing process broke down. The recursive version of the window process was more accurate but was very slow because of the multi-scale transforms required.

However, some means of combining the windowed Hough transform with the Least Squares vectorisation method, to provide a 'vector seeding' where the vectors discovered by the Hough transform can then be extended by the LMS model might prove to combine the best of both methods.

Within the test program simple list structures are employed to store the vector data - these are not well optimised for the tasks required. When comparing or combining vectors and building scan regions it is necessary to check each vector against every other entry because the list does not provide any spatial structure with respect to the vector location. A better method for storing the vector data would be to employ a quad-tree storage mechanism. Such a data structure would provide a more efficient search and compare method that may avoid unnecessary comparisons and therefore speed the process of combining vectors and building scan regions when using the 'nearest vector' method.

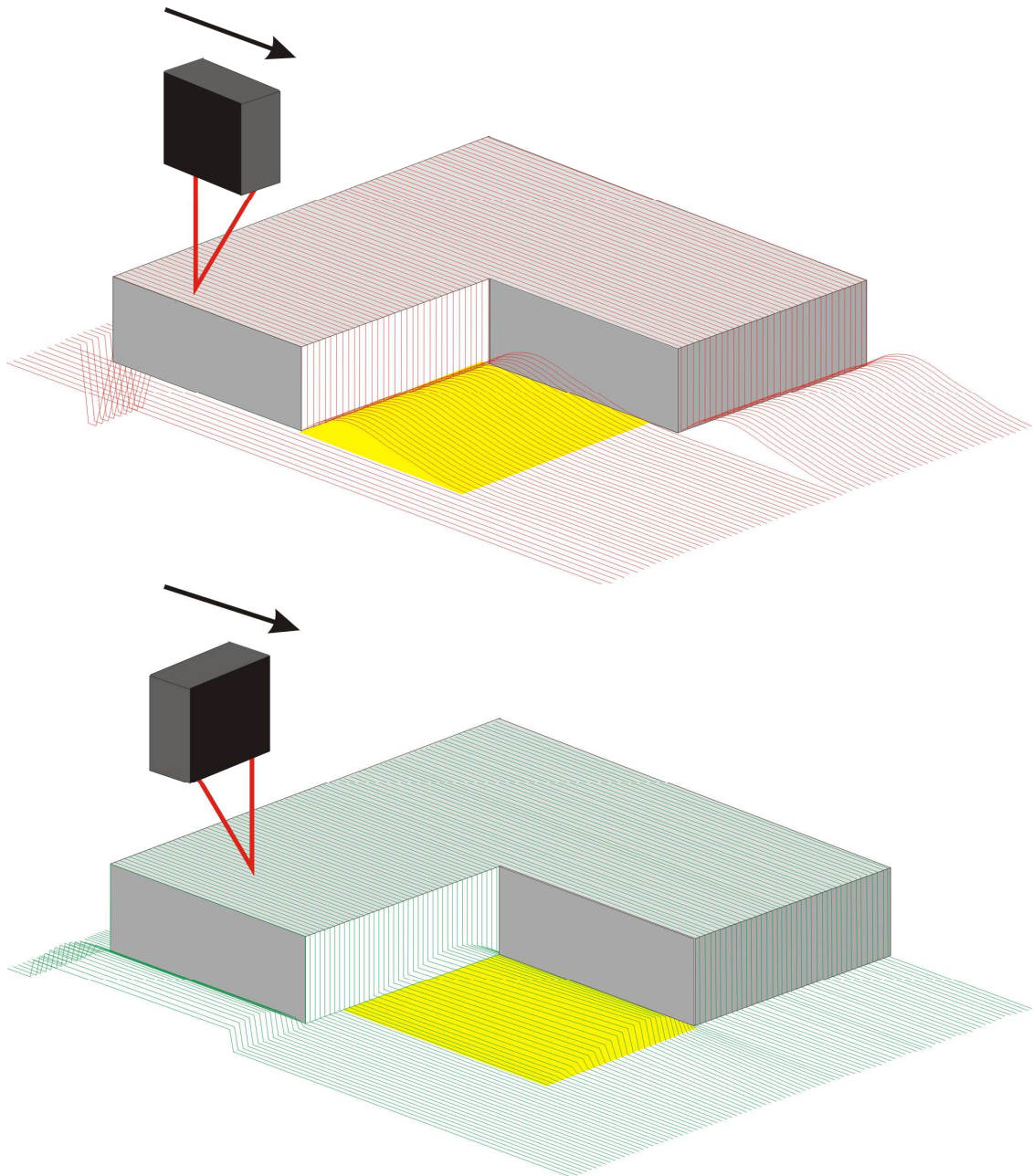
### **8.3 Compensation for 'Irreconcilable' Scan Data Points**

Some areas of the scan image, notably those in the proximity of 'internal corners' cannot be corrected using any of the algorithms suggested in this project. Even using Wong's method at these locations will not remove the reflection effect because, no matter what orientation the laser approaches the edge, there will always be some secondary reflection from one or more edges. Thus the data value is always distorted no matter what orientation the scan is made – even the 'best' orientation data contains some degree of distortion. This is shown in Figure 8-1.

One potential approach to compensating for this error would be to recognise where these positions exist (possibly using some form of corner feature detection in the 2D and / or 3D image) and then to extrapolate the data from a local region along the edge that is just outside the area of effect of the corner from a scan orientation that is parallel to the approaching edge and therefore contains minimal distortions. Sampling a number of values may be required to check for patterns in the data such as a regular 'ripple' effect in the surface.

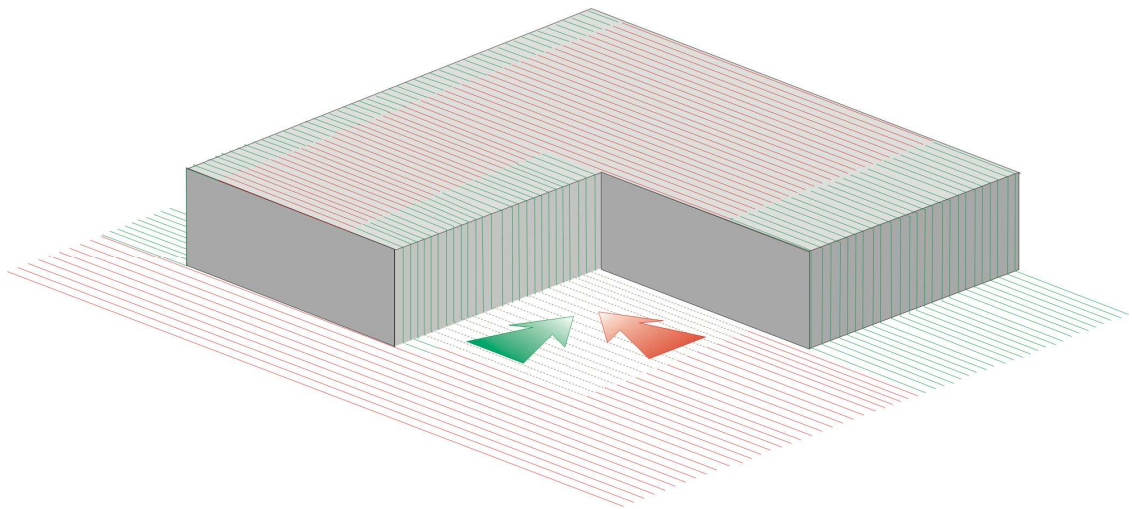
It may be that extrapolating the data does not fully represent the true contours of the area within the corner region. A better method may be to find the difference between the

orientation data we assume to be correct and the data which exhibits the bow wave error in the region outside the influence of the corner and then apply that difference to the data for that 'incorrect' orientation within the corner region.



**Figure 8-1A):** Data at an internal corner displays errors (within the yellow shaded regions) for any scan orientation as it is not possible to avoid reflections from vertical surfaces at any angle.





**Figure 8-1 B) In combining the above scans, extrapolating data from orientations parallel to the edge from outside the region affected by the internal corner may help to reduce the errors.**

## **8.4 Other Methods**

A number of other potential methods of improving the scan results have been considered and discussed during the development of this project. Some of these concepts are discussed in brief here.

### **8.4.1 Pre-Scan Profiling**

Although the edge detection analysis is able to find the position of edges in the image it does not provide any information about the relative height of the regions for which that edge marks the boundary.

One of the considerations in the orientation of the laser is whether the edges that the scanner is approaching are higher or lower than the current height – this is especially important in the case where ‘internal corners’ (conflicting edge orientations within the error radius of each other) are encountered.

It may be possible through the use of selected laser scan lines to provide a height profile across the image in a number of positions (and at opposing orientations). This would require some user input to determine the axes of these ‘pre-scan’ lines (as well as some care in selecting an optimal laser orientation that would not cause errors with respect to the geometric edges being scanned).

Although it would require that the end-user was aware of the causes of the distortions that this project has attempted to avoid, it is envisaged that the operator would be able to plot simple lines of cross-section onto the captured camera image at positions of interest, and then let the system determine the position of those lines in terms of scan coordinates and map the captured height profile information onto the image. This height information could then be used to help determine the best laser orientation with respect to approaching edges.

It is worth noting that there are two possible ‘correct’ (i.e. parallel) orientations of the laser for a single edge – either parallel with the emitter to the left and detector to the right or vice versa. However in the situation where we have conflicting edge orientations it is likely errors will be unavoidable for one or other of the edges. The pre-scan profiling may help to make decisions about which of the ‘optimal’ laser orientations will produce the least significant magnitude of errors. Another potentially useful benefit of such profiling would be that it may allow the width of scan regions (and therefore the total area scanned) to be varied with relative edge heights rather than assuming the worst for all regions.

#### **8.4.2 Material Reflectivity Index**

One of the difficulties in determining the presence of errors in the scan data is that the profile and extent of these errors varies with the relative reflectivity of the scan bed and the scan object and is not always possible to spray the scan object with a matte coating to provide a ‘neutral’ response.

There is some potential in investigating the laser response profiles of different materials under controlled conditions to determine a ‘material reflectivity index’ which may help in gauging the extents of scan regions required to avoid the occlusion and reflection errors.

Given the number of material types (and the even larger number of material combinations) that may be of interest this might prove to be a difficult task, however it might be sufficient to provide a number of sample template profiles that might aid in determining the likely response of the object under consideration in combination with some user decision of the template type that the object most closely resembles.

### **8.4.3 ‘Just in Time’ Edge Detection**

The method used in this project involves an image scene that encompasses the whole of the object (or at least a large percentage of the object area) that is pre-processed and all the edges in the scan area are determined ahead of the scan region development. However, in general, the edge effects are local to a small proportion of the image and therefore, for any given position in the scan, information is only required about a small area around the current scan point to determine the orientation of edges on a ‘local’ scale

An alternative method would be to employ a camera that covered only a small section of the object at any time. This camera would then be moved ahead of the laser scanner during the scan process, with the camera a set (known) distance ahead of the laser allowing enough time for the small area covered by the image scene to be processed by the required algorithms. This method would probably require dedicated image analysis hardware in order that it would be fast enough to process the image space ahead of the laser with enough time to orient the laser as required prior to the scanner arriving at the location represented in the image scene.

Because the area covered by the camera would be much smaller, the required image resolution might be reduced (and therefore the processing time for the image would be lower). Additionally, as the camera would be moving ahead of the scanning device, it would effectively be capturing a moving image where most of the contents from one frame to the next would be just shifted slightly according the movement of the camera, further reducing the required processing.

### **8.4.4 Mapping of image data to 3D Data Object (‘skinning’)**

The image data represents the object in 2 dimensions and the point cloud data represents the object in 3 dimensions, however the appearance of the point cloud is sometimes difficult to interpret. It may aid the operator if the image can be mapped over the point cloud data as a ‘skin’, using the height as an offset. This is not a new idea – it is commonly done with 3D models of facial data, where the subject is photographed and then the photo data is mapped onto the 3D model.

### **8.4.5 Further User Interaction**

One of the requisites for this project was to minimise the level of user interaction required to perform the scanning. Some selection of parameters is required for the image analysis process and currently only manual calibration methods are possible. Reducing the level of interaction may be possible, or at least simplifying the parameter selection with a step-by-step process.

In order to reduce the amount of image processing required it should be possible for the user to select a limited section of the image for processing. This is particularly useful where the total image area is larger than the area for which scan data is available, as the area outside the scan data is essentially redundant. In addition to reducing the processing workload, ignoring outlying areas can help with determining more precise thresholding levels as the redundant data values will not be factored into the calculations. Such selection could be done by ‘marquee-dragging’

It should be possible to let the operator override the selection of region orientations and provide their own region map. The image analysis is almost never ‘perfect’ and user-enhancements to the process may help to resolve troublesome areas. A drawing tablet and stylus might prove to be a useful input device, allowing the user to make adjustments to the image analysis process, or use the registered camera image in order to provide their own region orientation map.

### **8.4.6 Artificial Intelligence Approaches**

Further development of this idea could be to use some artificial intelligence approach to recognising the profiles and likely occurrence of occlusion and reflectance errors and to compensate for them automatically, although this is a completely different line of research. It may be possible to train a neural net to recognise the profile of an edge where a bow wave or occlusion error is present. It is likely that a large number of input variables would be required in order for this to work, such as knowledge of the current orientation relative to that edge and the relative reflectivity of the surfaces as well as the edge profile in at least two or three dimensions; but if the profile of the error can be recognised as such (and distinguished from a real edge with a similar profile) then potentially some automatic compensation can be applied to the data.

---

<sup>125</sup> Shu-Yu Zhu, Konstantinos N. Plataniotis, Anastasios N. Venetsanopoulos  
'Comprehensive Analysis of Edge Detection in Color Image Processing', *Optical Engineering*, Volume 38, Issue 4, pp. 612-625, April 1999

<sup>126</sup> M. Ruzon and C. Tomasi, "Color Edge Detection with the Compass Operator," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Ft. Collins, CO, V. 2, pp. 160-166, June 1999.

<sup>127</sup> M. Ruzon and C. Tomasi, "Corner Detection in Textured Color Images," In *7th IEEE International Conference on Computer Vision*, Kerkyra, Greece, V. 2, pp. 1039-1045, September 1999.

<sup>128</sup> M. Ruzon and C. Tomasi, "Edge, Junction, and Corner Detection Using Color Distributions," In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, V. 23, No. 11, pp. 1281-1295, November 2001.

<sup>129</sup> J. van de Weijer Th. Gevers J.M. Geusebroek, 'Color Edge Detection by Photometric Quasi-Invariants', *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003)*

## Chapter 9 Project Summary and Conclusions

Previous work in this Research Group determined that the orientation of a single-perspective triangulation sensor relative to the edges of the object under consideration was the primary cause of occlusion and secondary reflection errors in the resulting scan data. Wong<sup>1</sup> suggested a method by which compensation for these errors could be applied to the scan. However this method requires a set of (at least) eight scans to capture the object from a number of orientations in order to determine the difference between the resulting scan outputs. Wong's method produced good results in terms of the resulting scan quality but was prohibitive with respect to the scan time required to capture the object from the necessary orientations.

This project has attempted to use optical image data from commercially available inexpensive digital image cameras to plan the scan process when using a single-perspective triangulation laser sensor. The idea is use information from a 2D camera image prior to the scan in order to detect features (such as edges) that may cause problems. This information can be used to determine the optimal scan orientation with respect to local features of the object under consideration, thereby avoiding scan distortions.

In general the sensor orientation can be optimised at any particular location (i.e. keeping it as close to parallel to local edges) by controlling its rotation (around the z-axis) to avoid scan errors. This can be achieved by performing multiple partial scans of the object (with the sensor at a fixed orientation during each scan), or potentially by rotating the sensor orientation during a single-pass scan operation.

This appears to be a novel solution to a common problem. Current commercial products rely on human operators to change the sensor alignment when scans generate artefacts: i.e. the scan results are assessed and the object position is adjusted. Although many systems allow the merging of multiple scans from different views at a post-scanning stage no attempt appears to have been made (either commercially or in research literature) to use image analysis as a means of solving the problem of determining the optimal sensor orientation prior to the scan.

The first stage of the project was to develop useful information from the camera image. As the primary features responsible for errors are geometric edges the application of known edge detection methods was investigated.

To provide a fair comparison of a number of different edge detection algorithms in order to determine the most appropriate for the situation, a generalised filter kernel was developed by the Author into which the required filter kernels could be loaded. This would also prove useful for a commercial product as conditions might change such that a different method became optimal. New kernels or different methods of calculating the output can be plugged-in to the convolution algorithm with minimal effort in changing the software.

Most of the edge detection methods tested provided acceptable results. First derivative edge detection operators proved useful because pixel orientation information can be derived from the gradient of the edge. Second derivative operators do not provide such information directly and in many cases proved too sensitive to weak edges for reliable use. Initial investigation by the author determined that the Canny algorithm proved reliable across a wide variety of images, although it requires some degree of operator interaction in determining the required optimal parameter settings.

The algorithm for the thresholding stage of the Canny algorithm was developed by the Author, based on a recursive edge-tracking method that follows edges where gradient magnitude falls between the defined threshold values in order to determine whether a potential edge pixel is connected to a recognised edge via a chain of such mid-threshold pixels.

The edge detection used to determine the geometry of the image is extremely sensitive to variations in the image intensity. Smoothing the image can remove some of these ‘false’ edges but at the expense of some edge delocalisation, especially visible as rounding of corners, which are critical features within the scan as they are usually positions that display distortion errors at a number of orientations.

Vectorisation algorithms, based on known edge detection techniques, have been developed to determine the position of vectors corresponding to the discovered edges. Further algorithms have been developed to process these vectors into ‘scan regions’ corresponding

to each particular scanner orientation. When the object is scanned at the orientation corresponding to the scan region the distortions are likely to be much reduced.

The work done has demonstrated the feasibility of the multiple scan approach once camera distortion, calibration and parallax issues have been dealt with. Two approaches were investigated by which partial scan orientation maps might be produced based on algorithms developed by the Author.

An initial ‘first pass’ solution provides a best single orientation scan based on the image analysis by determining the total edge length per orientation range<sup>i</sup> (as defined by the system operator). This allows the total number of errors to be minimised and could be used to begin the scan acquisition process (if one orientation proved to be dominant) whilst a more complex method of analysis is processed. The results from the complex analysis could then be merged with the single best scan to provide a composite output. Partial scan methods are also compared against this method.

The first ‘partial scan’ method (Scan Region Orientation) expands scan regions of a fixed size<sup>ii</sup> around vectors discovered by the image analysis process. This method was quick to generate an output map, however it was found to be prone to errors where regions representing conflicting orientations overlap. This is especially true at corners where the scan region method proved poor in eliminating errors. This method could be improved by reducing the areas of region overlap to a minimum and a better means of resolving data within remaining areas of region overlap.

The second method (Nearest Vector Orientation) seeks to avoid region overlap by determining the optimal orientation for each scan point based on the orientation of only the closest edge vector. This method produced better results although the processing time was considerably longer (especially when there are many vectors in close proximity); however the processing time was still much shorter than the actual task of scanning the object under consideration. This method is also sensitive to calibration issues, leading to potential misalignment of image regions with the scan file, hence the occurrence of obvious errors in

---

i The orientation range is the quantisation of contiguous angles represented by a single value within the system. Ideally this should be equal to or less than the orientation tolerance (see Section 4.1)

ii The size of the scan region is ‘fixed’ by the calibration information with respect to sensor height, triangulation angle and the maximum edge height.



continuity where the incorrect orientation file was selected. Although the scan region method also suffered from this, the averaging of overlapping regions disguises this effect to some extent.

It would be possible to combine the advantages of both methods: i.e. the Scan Region Orientation method could be applied to areas of the scan where there is a clear choice for the optimal orientation, and the Nearest Vector Orientation could then be used to resolve just the areas where the scan regions overlap (thus reducing the processing time of the second method).

Both the ‘partial scan’ methods that were implemented for testing are quite successful at reducing occlusion and bow wave errors along edges where the curvature is reasonably low (with respect to the scale of the image). With correct calibration and parameter settings the methods show 100% removal of edge artefacts along straight, uninterrupted lines away from region boundaries.

For research purposes the object scans were performed prior to the analysis process. Whilst this required that full-object scans were provided from a number of pre-defined orientations this method allowed these scans to be combined as a ‘virtual’ scan path output, thereby allowing many trials of different algorithms for combining partial scan areas. A simple change (such as variation of parameter settings) could therefore be tested without requiring the scan to be re-acquired in order to determine the results. This was useful because the scan acquisition process takes many times longer than the software analysis, although it does not always allow the errors to be reduced as much as if the scan orientations were selected to match the discovered edge orientations. However in the commercial version, the aim is to perform only the partial scans at determined orientations. It may be possible to provide a scan path to re-orient the head as the scan proceeds, thus minimising the overall scan time while still substantially reducing artefacts.

Some problems remain which are inherent to the nature of the laser triangulation method. Internal corners and small holes in the object’s surface have local regions where all orientations of the sensor will display significant distortions. The image analysis process allows these areas to be flagged for attention in a post-scan processing stage.

The feasibility of the single-pass scan using a continually variable sensor orientation is supported by the results, however further investigation is required to resolve a number of practical problems. In particular there may be some areas of objects where the sensor orientation could not be rotated fast enough to match the optimal orientation (e.g. when a sudden change between conflicting edge orientations occurs), unless the movement of the sensor was slowed or stopped whilst the rotation was adjusted. The architecture of the sensor data acquisition process does not currently allow this to occur. Such regions (where the edge orientation changes rapidly between conflicting orientations) may require areas to be rescanned at a locally optimal orientation.

The partial scan by orientation algorithms presented as novel methods in this thesis provide an advantage over Wong's method. Wong required 8 complete scans of the object and was thus limited by time, but also the alignment between an edge and the closest scan orientation could be as much as  $\pm 22.5^\circ$  (as Wong used a  $45^\circ$  orientation separation).

Testing during the early part of this project showed that errors could be perceived with a difference in alignment between scan and edge orientation of as little as approximately  $\pm 10^\circ$ . The method presented here is limited in the accuracy of alignment only by the granularity of orientation selected for the binning process (which impacts the processing time and memory used). A commercial scan head control would use a stepper motor and could achieve very small angle bins. Of course, frequent small changes of scan head angle would slow the scan process, so a working compromise would have to be reached based on the required scan accuracy and the time to scan the object.

Further work is required in determining the difference between detected edges that represent features that may cause scanning errors and edges that are generated as a result of noise or texture in the image. These 'noise' lines may cause the selection of an incorrect orientation and increase the processing overheads in all cases. Although texture (and also 2D transitional edges caused by surface patterning) is of some concern, the edges generated by these features have less effect on the overall quality of output than the geometric edges and may cause a local mis-selection of orientation. A combination of image analysis techniques, with some AI method of developing a 'consensual output' may be possible in order to determine which edges are the most important.

The largest remaining problem with respect to this work is the accurate calibration of the system, specifically the calibration of camera alignment and corrections for parallax error and lens aberrations. Algorithms have been developed to compensate for these effects. The quality of the imaging device used has a large impact on the quality of the resulting output. Good quality camera optics are required to provide a clearly focused image with the minimum of distortions. The webcam tested during this project fell short of the expected standards due to the poor quality of the camera optics. The digital camera performed well at even a low image resolution<sup>iii</sup>. As the cost of good quality digital imaging devices continues to fall there is a wide choice of reasonably-priced hardware suitable for the task. A commercial product based on these methods would have to be well-engineered to ensure calibration and alignment between the sensor and camera images (see Chapter 4.3).

Despite the many issues that remain with accurate calibration and the problems of determining the necessary parameters for clean edge detection of ‘real world’ objects the initial hypothesis that image analysis may be used as guide to correct orientation of the laser sensor is borne out by the experiments.

The research conducted so far on this project contributes significant progress towards the goal of using image analysis as a means of identifying regions of the scan likely to be prone to errors and the avoidance of such distortions by the automatic selection of a locally optimal scan orientation, however there is still much work to be done before these methods will be useful in a commercial context. Further investigation is required particularly with respect to the resolution of areas of conflicting orientation and the accurate transformation between the image and scan coordinates.

---

<sup>iii</sup> There is a relationship between the required image resolution and the intended scan resolution but this was not explored in this project as the 640x480 image was adequate for the coarse scan resolution employed.

## Table of References

- [1] K. Wong, '*Compensation for Distortion in the Imaging Process for 3D Surfaces*', PhD. Thesis, The Nottingham Trent University, 2002.
- [2] Toubin M., Laligant O., *et al.*, '*Multiscale Analysis for the Characterization of 3-D Objects*', Proceedings of SPIE, Intelligent Robots and Computer Vision XVII: Algorithms, Techniques and Active Vision, vol. 3522, pp.414-423, October 1998.
- [3] Karbacher S. and Haeusler G., '*New Approach for the Modelling and Smoothing of Scattered 3-D Data*', Proceedings of SPIE, Three-Dimensional Image Capture and Applications, vol. 3313, pp.168-177, March 1998.
- [4] Shu C. and Xi F., '*Model-based Scanning Path Generation for Inspection*', Proceedings of Second International Conference on 3-D Digital Imaging and Modeling (3DIM '99), pp.118-124, Ottawa, Canada, October 1999.
- [5] Lee, S.J. *et al.* '*Laser Scanning Probe With Multiple Detectors Used For Sculptured Surface Digitization In Reverse Engineering*' 2005 J. Phys.: Conf. Ser. 13 pp.155-158
- [6] Lynch M., '*Computer Numerical Controls for Machining*', McGraw-Hill Professional Publishing, 1992.
- [7] Ingle K. A., '*Reverse Engineering*', McGraw-Hill Professional Publishing, 1994.
- [8] J.C. Owen, P.J. Sloan, and W.B. Thompson, '*Interactive Feature Based Reverse Engineering of Mechanical Parts*', Proceedings of the ARPA Image Understanding Workshop, November, 1994
- [9] Karbacher S., Babst J., and Häusler G., '*Visualization and Detection of Small Defects on Car-Bodies*', Vision, Modeling and Visualization '99, pp. 1-8, Sankt Augustin, Germany, 1999.
- [10] Lassiter J., '*Principles of Traditional Animation Applied to 3-D Computer Animation*', Proceedings of SIGGRAPH '87, pp. 35-44, July 1987.
- [11] Levoy M., '*The Digital Michelangelo Project*', Proceedings of Second International Conference on 3-D Digital Imaging and Modelling (3DIM '99), pp. 2-11, Ottawa, Canada, October 1999.
- [12] Várady T., Martin R. R. and Cox J., '*Reverse Engineering of Geometric Models – An Introduction*', Journal of Computer-Aided Design, vol. 29, no. 4, pp. 255-268, 1997.
- [13] J.-A. Beraldin, F. Blais, L. Cournoyer, M. Rioux, S.H. El-Hakim, R. Rodella, F. Bernier and N. Harrison, '*3D Digital Imaging and Modeling on Remote Sites*', Proc. Second Intern.Conf. on 3D Digital Imaging and Modelling, Ottawa, Canada, (1999).
- [14] R. A. Jarvis, '*A Perspective on Range Finding Techniques for Computer Vision*', IEEE PAMI, Vol. 5, No. 2, 1983 pp 122-139
- [15] Besl, P.J. '*Active, optical range imaging sensors.*' Machine Vision and Applications (1988) vol.1, no.2, p. 127-52.
- [16] H. J. Tiziani, '*Optical metrology of engineering surfaces-scope and trends*' in Optical Measurement Techniques and Applications, P. K. Rastogi, Ed., Artech House, Boston (1997).

- [17] F. Chen, G.M. Brown, M. Song, '*Overview of three-dimensional shape measurement using optical methods*', Opt. Eng., 39, pp. 10-22, Jan. 2000.
- [18] Blais, F. '*Control of low inertia galvanometers for high precision laser scanning systems*', Opt. Eng. 27(2): pp. 104-110; 1988.
- [19] Blais, F. '*Review of 20 Years of Range Sensor Development*', Journal of Electronic Imaging Vol.13 No.1. pp 231-240, Jan. 2004
- [20] Allen P. K. and Michelman P., '*Acquisition and Interpretation of 3-D Sensor Data from Touch*', IEEE Transactions on Robotics and Automation, vol. 6, no. 4, pp. 397-404, August 1990.
- [21] Kak, A. & Slaney, M., '*Principles of Computerized Tomographic Imaging*', 1988, IEEE Press, ISBN 0-87942-198-3
- [22] Trifonov, B, Bradley, D. & Heidrich, W. '*Tomographic Reconstruction of Transparent Objects*', ACM Siggraph 2006, Boston, MA, ISBN 1-59593-366-2
- [23] Fisher P. F. and Lindenber R. E., '*Distinctions among Cartography, Remote Sensing and Geographic Information Systems*', Photogrammetric Engineering and Remote Sensing, vol. 55, no. 10, pp. 1431-1434, 1989.
- [24] Jansa J., Huang Y. R. & Trinder J. C., '*Problems of Precise Target Location and Camera Orientation in Digital Close-Range Photogrammetry*', Proceedings of SPIE, Videometrics II, vol. 2067, pp. 151-161, October 1993.
- [25] Fraser C. S., '*Photogrammetric measurement to one part in a million*', Photogramm. Eng. Remote Sens. vol.58 No.3, pp.305-310 (1992).
- [26] Nayar, S., Watanabe, M. & Noguchi, M. '*Real-time Focus Range Sensor*', Proceedings of IEEE International Conference on Computer Vision, pp. 995-1001, 1995.
- [27] Koskinen M., Kostamovaara J. T. & Myllyla R. A., '*Comparison of Continuous-Wave and Pulsed Time-of-Flight Laser Range-Finding Techniques*', Proceedings of SPIE, Optics, Illumination and Image Sensing for Machine Vision VI, vol. 1614, pp. 296-305, March 1992.
- [28] Moody S. E., '*Commercial Applications of Lidar: Review and Outlook*', Proceedings of SPIE, Optical Remote Sensing for Industry and Environmental Monitoring, vol. 3504, pp. 41-44, August 1998.
- [29] Moring I., Ailisto H., Koivunen V., & Myllyla, R. '*Active 3-D vision system for automatic model-based shape inspection*', Opt. Lasers Eng. 10, pp. 3-4, 1989.
- [30] Lihachev D. I., '*Prospects of Laser Interferometers*', Proceedings of SPIE, International Conference on Lasers for Measurements and Information Transfer, vol. 4316, pp. 43-45, November 2000.
- [31] Michelson A. A., '*On the Application of Interference Methods to Astronomical Measurements*', Philosophical Magazine and Journal of Science, vol. 30, pp. 1-20, July 1890.
- [32] Xue M., Xiangli B. and An B., '*Optical Systems of Imaging Interferometers*', Proceedings of SPIE, International Optical Design Conference, vol. 3482, pp. 474-483, September 1998.
- [33] Curless, B. '*New Methods for Surface Reconstruction from Range Images*'. Thesis published 1997, Stanford University, USA

- [34] Optonor website <http://www.optonor.no/SNT.htm> (accessed 30/08/06)
- [35] K. Harding and L. Bieman, '*High speed moiré contouring methods analysis*', Proc. SPIE 3520, pp. 27-35 (1998).
- [36] Harthong J., Becker A., '*Inverse Moiré*', Proceedings of SPIE, Optical Inspection and Micromeasurements II, vol. 3098, pp. 2-9, September 1997.
- [37] Altschuler M. D., Altschuler B. R. and Taboda J., '*Laser Electro-Optic System for Rapid 3-D Topographic Mapping of Surfaces*', Optical Engineering, vol. 20, no. 6, pp. 953-961, December 1981.
- [38] Sato K. and Inokuchi S., '*Three-Dimensional Surface Measurement by Space Encoding Range Imaging*', Journal of Robotic Systems, vol. 2, no. 1, pp. 27-39, 1985.
- [39] Reichmann W., '*Fast Object Recording by means of Structured Light and Photogrammetric Techniques*', Proceedings of IAPRS, vol. 30, pp. 195-200, 1995.
- [40] Srinivasan V., Liu H. C. and Halioua M., '*Automated Phase-Measuring Profilometry: A Phase Mapping Approach*', Applied Optics, vol. 24, no. 2, pp. 185-188, 1985.
- [41] Fang Q. and Zheng S., '*Linearly Coded Profilometry*', Applied Optics, vol. 36, no. 11, pp. 2401-2407, 1997.
- [42] Gaertner H., Lehle P., Tiziani H. J. & Volland C., '*Structured Light Measurement by Double Scan Technique*', Proceedings of SPIE, Vision Systems: Sensors, Sensor Systems and Components, vol. 2784, pp. 21-30, August 1996.
- [43] Keferstein C. P., and Marxer, M. '*Testing Bench for Laser Triangulation Sensors*', Sens. Rev. vol 18 no. 3, pp. 183-187. 1998
- [44] W. J. Smith, *Optical Engineering*, 2nd ed., McGraw-Hill, New York, 1990
- [45] Idesawa, M. '*High-precision image position sensing methods suitable for 3-D measurement*', Opt. Lasers Eng. 10, 3-4 (1989).
- [46] Keyence Technical Report on Sensors and Measuring Instruments (1997).
- [47] S. Motavalli, B. Bidanda, '*A Part Image Reconstruction System for Reverse Engineering of Design Modifications*', Journal of Manufacturing Systems , Vol. 10, No. 5, 1991, pp 383-395
- [48] Modjarrad, A. '*Non-Contact Measurement Using a Laser Scanning Probe*', SPIE, Vol. 1012 In-Process Optical Measurements, 1988, pp 229-239
- [49] Rioux M., '*Laser Range Finder based on Synchronised Scanners*', Applied Optics, vol. 23, no. 21, pp. 3837-3843, 1984.
- [50] de Bakker M., Verbeek P. W. & Steenvoorden G. K., '*Smart PSD Array for Sheet-of-Light Range Imaging*', Proceedings of SPIE, vol. 3965, pp. 21-32, May 2000.
- [51] Häusler G. & Herrmann J., '*Physical Limits of 3-D Sensing*', Proceedings of SPIE, Optics, Illumination and Image Sensing for Machine Vision VII, vol. 1822, pp. 150-158, March 1993.
- [52] Häusler G., '*Three-Dimensional Sensors – Potentials and Limitations*', Handbook of Computer Vision and Applications, Academic Press, vol. 1, no. 19, pp. 485-506, 1999.
- [53] Wolff L. B., '*Relative Brightness of Specular and Diffuse Reflection*', Optical Engineering, vol. 33, no. 1, pp. 285-293, January 1994.

- [54] Dorsch R., Herrmann J. and Häusler G., '*Laser Triangulation: Fundamental Uncertainty of Distance Measurement*', Applied Optics, vol. 33, no. 7, pp. 1306-1314, 1994.
- [55] Shu C. and Xi F., '*Model-based Scanning Path Generation for Inspection*', Proceedings of Second International Conference on 3-D Digital Imaging and Modeling (3DIM '99), pp. 118-124, Ottawa, Canada, October 1999
- [56] Amann M. C., Bosch T. M., et al., '*Laser Ranging: A Critical Review of Unusual Techniques for Distance Measurement*', Optical Engineering, vol. 40, no. 1, January 2001.
- [57] Ballard, D. & Brown, C. '*Computer Vision*' Prentice Hall Englewood Cliffs, NJ. 1982
- [58] Gonzales & Woods '*Digital Image Processing*' Prentice Hall; 2nd International Ed edition, 2003 ISBN: 978-0130946508
- [59] <http://www.cs.cmu.edu/~cil/vision.html> (last accessed 23-3-07).
- [60] Fu, K.S. & Mui, J.K., '*A Survey On Image Segmentation*', Pattern Recognition, vol. 13, pp. 3-16, 1981.
- [61] Low, A. '*Introductory Computer Vision and Image Processing*' McGraw-Hill, 1991, Chapter 6, p.69.
- [62] Pitas, I. and Venetsanopoulos, A.N., '*Edge Detector Based on Non-Linear Filters*', IEEE Transactions on Pattern Matching and Machine Intelligence, Vol.8 no.4 pp 538-550, 1986
- [63] <http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm> (last accessed 10 October 2006)
- [64] Poggio, T. & Torre, V. '*On Edge Detection*', IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8 no.2, pp.147-163, March 1986.
- [65] Jain, A.K., '*Fundamentals of Digital Image Processing*', Prentice Hall, 1989.
- [66] Nalwa, V.S., '*A Guided Tour of Computer Vision*', Addison-Wesley, 1993
- [67] Zamperoni, P. '*Image Enhancement*', Advances in Imaging and Electron Physics, Academic Press, San Diego, 1995, pp. 1-76
- [68] Russo, F., '*Edge Detection in Noisy Images Using Fuzzy Reasoning*' IEEE Trans. Instrum. Meas. 47, 1998, pp. 1102-1105.
- [69] Kundu, A., '*Robust Edge Detection*' Pattern Recognition 23 pp. 423-440
- [70] Bovik A., Huang, T. & Munson, D. '*Non Parametric Tests for Edge Detection in Noise*', Pattern Recognition, Vol: 19 No: 3, Page(s): 209-219, May 1986
- [71] Ziou, D & Tabbone, S., '*Edge Detection Techniques - An Overview*', Pattern Recognition and Image Analysis, 8(4) Dec. 1998, pp. 537-559.
- [72] Lyvers, E.P., and Mitchell, O.R. '*Precision Edge Contrast and Orientation Estimation*', IEEE. Transactions on Pattern Analysis and Machine Intelligence, Vol.10 no.6 1988, pp.927-937.

- [73] Prewitt, J.M.S. '*Object Enhancement and Extraction*', in Picture Processing and Psychopictures, Eds. B. S. Lipkin and A. Rosenfeld, Academic Press, New York (1970).
- [74] Canny, J. F., '*A Computational Approach to Edge Detection*', IEEE Trans. Pattern Analysis & Machine Intelligence, Vol. 8, No. 6, 1986. pp. 679-698.
- [75] Deriche, R., '*Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector*', The International Journal of Computer Vision, Vol.1 no.2 1987, pp. 167-187
- [76] <http://www.cee.hw.ac.uk/hipr/html/canny.html> (last accessed 10 October 2006)
- [77] Marr, D. and Hildreth, E. C., '*Theory of Edge Detection*', Proc. Roy. Soc. London, 207 pp.187–217 Nov.1980
- [78] Marr, D. '*Vision*' Freeman, 1982, Chapter 2, pp 54-78
- [79] Heath, M., Sarkar, S., Sanocki, T. & Bowyer, K. '*Comparison of Edge Detectors, A Methodology and Initial Study*', Computer Vision and Image Understanding, Vol.69 No.1, Jan 1998. pp 38-54. (Article No. IV960587)
- [80] Shin M., Goldgof, D. & Bowyer, K. '*Comparison of Edge Detector Performance Through Use in an Object Recognition Task*', Computer Vision and Image Understanding, Vol. 84 No.1 pp. 160-178, Oct 2001
- [81] Kanaugo T., Jaisimha, M., Palmer J. & Haralick, R. '*A Methodology for Quantitative Performance Evaluation of Edge Detection Algorithms*', IEEE Trans. Image Processing, Vol. 4 no.12, pp. 1667-1673, Dec 1995.
- [82] M. Heath , S. Sarkar, T., Sanocki, & K. Bowyer, '*A Robust Visual Method for Assessing the Relative Performance of Edge Detection Algorithms*', IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.19, no.12, pp 1338-1359, Dec 1997
- [83] Freeman, H. & Davis, L. S. '*A Corner Finding Algorithm for Chain Code Curves*', IEEE Transactions on Computers, 26, pp.297–303 (1977).
- [84] Beaudet, P. R. '*Rotational Invariant Image Operators*', Int. Conference on Pattern Recognition, pp.579–583. 1978.
- [85] Moravec, H. P., '*Visual Mapping by a Robot Rover*', Joint Conference on Artificial Intelligence, pp.598–600, 1979.
- [86] Harris, C. G. & Stephens, M. '*A Combined Corner and Edge Detector*', 4th Alvey Vision Conference, pp.147–151 1988.
- [87] Zheng, Z., Wang, H. & Teoh, E. '*Analysis of Gray Level Corner Detection*', Pattern Recognition Letters, 20, pp.149 162 (1999).
- [88] Kitchen, L. & Rosenfeld, A. '*Gray-level Corner Detection*', Pattern Recognition Letters, 1, pp.95–102 (1982).



- [89] Wang, S.J. & Binford, T.O. '*Generic, Model-based Estimation and Detection of Discontinuities in Image Surfaces*'. In ARPA IUW, volume II, pp.113-116, 1994.
- [90] Smith, S. M. & Brady, J. M., '*SUSAN - a new approach to low level image processing*', International Journal of Computer Vision, 23(1), pp.45-78 (1997)
- [91] Sezgin, M. & Sankur, B., '*Survey Over Image Thresholding and Quantitative Performance Evaluation*', Journal of Electronic Imaging Vol 13. no.1 January 2004, pp.146-165
- [92] Adams, R. & Bischof, L., '*Seeded Region Growing*', IEEE Trans. on Pattern Analysis and Machine Intelligence, vol 16, no. 6, pp. 641-647, June 1994
- [93] Z. Lin, J. Jin, and H. Talbot, '*Unseeded region growing for 3d image segmentation*,' in ACM International Conference Proceeding Series, Selected papers from the Pan-Sydney workshop on Visualisation, vol. 2, 2000, pp. 31-37.
- [95] Ohlander, R., Price, K., & Reddy, D., '*Picture Segmentation Using a Recursive Region Splitting Method*', Computer Graphics and Image Processing, volume 8, 1978 pp. 313-333
- [96] Chen, P. C. & Pavlidis, T. '*Image Segmentation as an Estimation Problem*'. In Image Modelling, pp.9-28. Ed. A. Rosenfeld, Academic Press, 1981.
- [97] Spann, M. & Wilson, R. G. '*A Quad-Tree Approach to Image Segmentation Which Combines Statistical and Spatial Information*'. Pattern Recognition, 18(3/4):257-269, 1985.
- [98] Haralick, R. Shanmugam, K. & Dinstein I., '*Texture Features for Image Classification*'. IEEE Transactions on Systems, Man, and Cybernetics, SMC- 3(6): pp.610-621, 1973
- [99] Tombre, K. '*Analysis of engineering drawing: state of the art and challenges*' Graphics Recognition - Algorithms and Systems. K. Tombre, and A.K. Chhabra (eds), Springer-Verlag, Berlin, Germany, 1998. , pp. 257-264.
- [100] Tombre, K. & Tabbone, S. '*Vectorization in Graphics Recognition: To Thin or Not To Thin*'. Proceedings of ICPR'00. Vol.2, pp. 91-96, 2000.
- [101] Freeman, H., '*Boundary encoding and processing*,' in Picture Processing and Psychopictures. New York: Academic, 1970.
- [102] Hori O. & Tanigawa. S. '*Raster-to Vector Conversion by Line Fitting Based on Contours and Skeletons*' Proceedings of ICDAR'93, pp. 353-358, Tsukuba, Japan, 1993.
- [103] Ballard, D. H., '*Generalizing the Hough Transform to Detect Arbitrary Shapes*' Pattern Recognition, Vol.13, No.2, 1981. pp. 111-122
- [104] T. '*Hough Transform for Line Recognition: Complexity of Evidence Accumulation and Cluster Detection*.' CVGIP, 46(3): pp. 327-345, 1989.

- [105] Ji, Q. & Xie, Y. '*Randomised Hough Transform with Error Propagation for Line and Circle Detection*', Pattern Anal Applic., Vol 6. pp. 55-64, 2003
- [106] Maekynen A. J., Rahkonen T. E. and Kostamovaara J. T., '*CMOS Binary Position-Sensitive Photodetector (PSD) Array*', Proceedings of SPIE, Sensors, Sensor Systems and Sensor Data Processing, vol. 3100, pp. 101-109, September 1997.
- [107] Xu Y., Tang J. and Zhong W., '*New Method of Processing the Signals of a Position Sensitive Detector*', Proceedings of SPIE, Advanced Photonic Sensors: Technology and Applications, vol. 4220, pp. 260-263, October 2000.
- [108] From Matsushita Technical Document available at [www.matsushita.es/download/sensor/LM10\\_cat\\_en.pdf](http://www.matsushita.es/download/sensor/LM10_cat_en.pdf)
- [109] Amann M. C., Bosch T. M., et al., '*Laser Ranging: A Critical Review of Unusual Techniques for Distance Measurement*', Optical Engineering, vol. 40, no. 1, January 2001.
- [110] Poliakoff, J.F., Private communication to Author. 2006 (See Appendix A)
- [111] Tsai, R., '*A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf TV cameras and lenses*'. IEEE Journal of Robotics and Automation, 3(4):323-344, August 1987.
- [112] Faugeras O.D., Luong Q.-T., and Maybank S.J. '*Camera self-calibration: theory and experiments*'. In European Conference on Computer Vision, pp 321-34., 1992.
- [113] Faugeras, Olivier. '*Three-Dimensional Computer Vision*', MIT Press, 1993.
- [114] Tordoff, B.J. & Murray, D.W., '*The impact of radial distortion on the self calibration of rotating cameras*', Computer Vision and Image Understanding 96 pp.17-34 (2004)
- [115] Foley, J.D. et al. '*Introduction to Computer Graphics*' Ch. 5, pp.168-177, Addison Wesley, MA, 1994
- [116] Ruzon M. & Tomasi, C. '*Color Edge Detection with the Compass Operator*', In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Ft. Collins, CO, V. 2, pp. 160-166, June 1999
- [117] Novak, C.L. & Shafer, S.A. '*Color Edge Detection*', Proceedings DARPA Image Understanding Workshop, Vol. I, pp. 35-37, Los Angeles, CA, USA, February 1987.
- [118] [www.paintlib.de](http://www.paintlib.de). Last accessed 5/9/06
- [119] National Center for Manufacturing Sciences, '*The Next Generation Controller Part Programming Functional Specification (RS-274/NGC)*', Draft; NCMS; August 1994
- [120] Chai, I. and Dori, D., '*Orthogonal Zig-Zag: An Efficient Method for Extracting Lines From Engineering Drawings*', C. Arcelli, L. P. Cordella and G. Saruiti di Baja, eds., Visual Form. Nev York: Plenum Press, pp. 127-136, 1992,
- [121] <http://homepages.inf.ed.ac.uk/rbf/HIPR2/skeleton.htm>
- [122] R. Cucchiara, C. Grana, A. Prati, R. Vezzani, '*A Hough Transform-Based Method for Radial Lens Distortion Correction*', ICIAP, 12th International Conference on Image Analysis and Processing (ICIAP'03), p.182, 2003.
- [123] Satoru Yoneyama, Hisao Kikuta, Akikazu Kitagawa, and Koji Kitamura, '*Lens Distortion Correction for Digital Image Correlation by Measuring Rigid Body Displacement*', Opt. Eng. Vol. 45, Issue 2, Feb. 2006

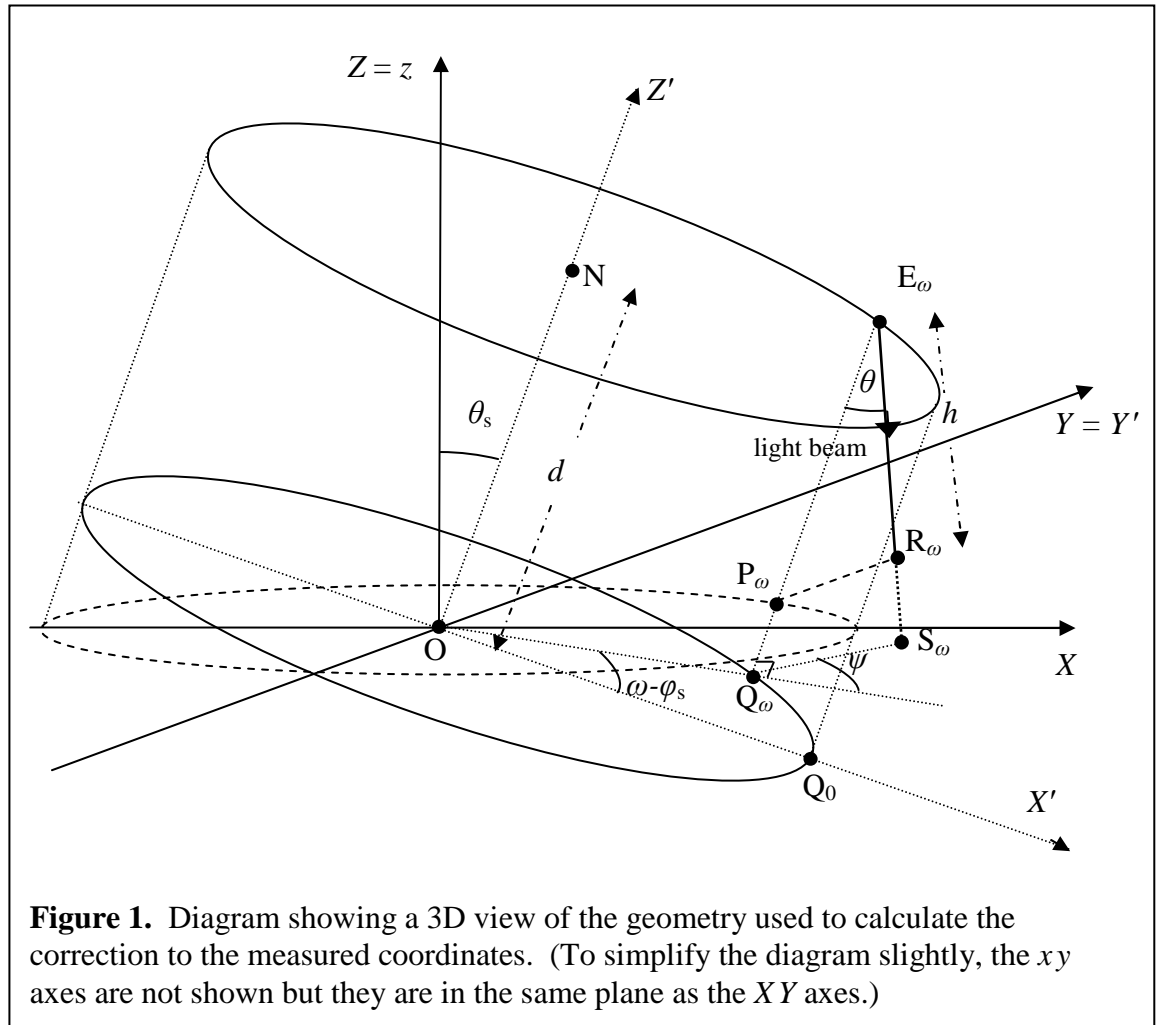
- [124] Toru Tamaki, Tsuyoshi Yamamura, Noboru Ohnishi, '*Unified Approach to Image Distortion*,' *ICPR*, p. 20584, 16th International Conference on Pattern Recognition (ICPR'02) - Volume 2, 2002.
- [125] Shu-Yu Zhu, Konstantinos N. Plataniotis, Anastasios N. Venetsanopoulos '*Comprehensive Analysis of Edge Detection in Color Image Processing*', *Optical Engineering*, Volume 38, Issue 4, pp. 612-625, April 1999
- [126] M. Ruzon and C. Tomasi, '*Color Edge Detection with the Compass Operator*,' In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Ft. Collins, CO, V. 2, pp. 160-166, June 1999.
- [127] M. Ruzon and C. Tomasi, '*Corner Detection in Textured Color Images*,' In *7th IEEE International Conference on Computer Vision*, Kerkyra, Greece, V. 2, pp. 1039-1045, September 1999.
- [128] M. Ruzon and C. Tomasi, '*Edge, Junction, and Corner Detection Using Color Distributions*,' In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, V. 23, No. 11, pp. 1281-1295, November 2001.
- [129] J. van de Weijer Th. Gevers J.M. Geusebroek, '*Color Edge Detection by Photometric Quasi-Invariants*', *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003)*

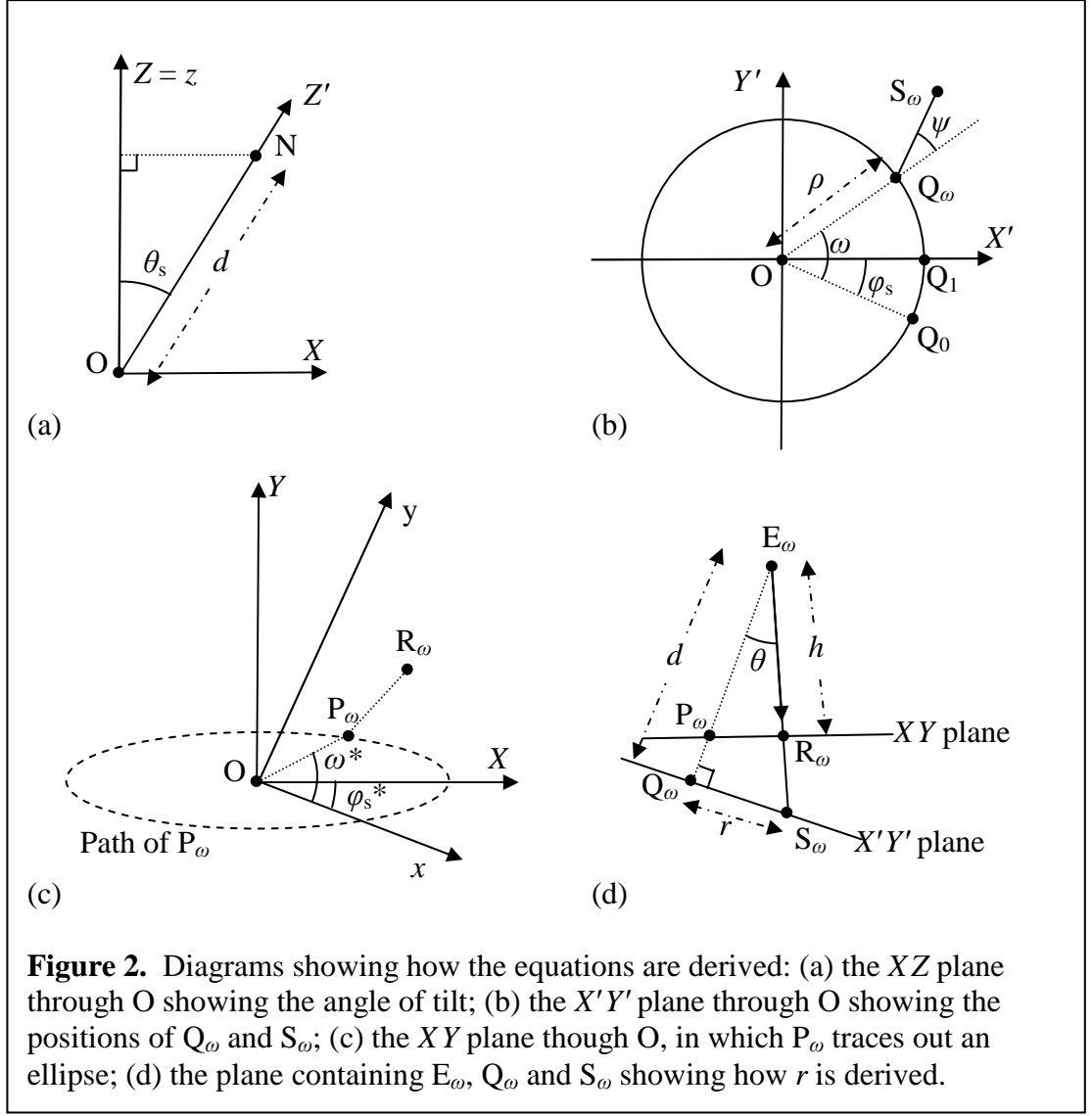
## Appendix A - Derivation of the Correction Algorithm for Laser Calibration

We assume that the  $z$  axis of the CNC machine is vertical. Suppose the angle of tilt of the axis of rotation of the scanner from the vertical is  $\theta_s$ . Then, if the light beam is parallel to that axis, the point where the beam falls will describe an ellipse on a horizontal plane, as indicated by the dashed line in Figure 1. We assume in the general case that the beam is not parallel but tilted relative to the axis by an angle  $\theta$ .

Consider the situation when the scanner has been rotated through an angle  $\omega$  and let  $x_m$ ,  $y_m$  and  $z_m$  be the measured values from the scanner. We are seeking the corrected values, which we shall call  $x^*$ ,  $y^*$  and  $z^*$ .

As shown in Figure 1, we define  $N$  as the centre of rotation of the emitter about the axis of rotation,  $E_\omega$  as the position of the emitter for angle  $\omega$  and  $\rho$  as the radius of





**Figure 2.** Diagrams showing how the equations are derived: (a) the  $XZ$  plane through  $O$  showing the angle of tilt; (b) the  $X'Y'$  plane through  $O$  showing the positions of  $Q_\omega$  and  $S_\omega$ ; (c) the  $XY$  plane through  $O$ , in which  $P_\omega$  traces out an ellipse; (d) the plane containing  $E_\omega$ ,  $Q_\omega$  and  $S_\omega$  showing how  $r$  is derived.

the rotation. Now suppose that the spot where the beam hits the object is  $R_\omega$  and take the origin  $O$  to be the point where the axis of rotation hits the horizontal plane through  $R_\omega$ . Now  $x$ ,  $y$  and  $z$  are the axes of the CNC machine but, to simplify the derivation, we choose new axes  $Z = z$  and  $X$  and  $Y$  which lie in the  $xy$  plane such that the axis of rotation is tilting from the  $z$  axis towards the  $X$  axis, as shown also in Figures 1 and 2(a).

Next we define a third set of axes such that  $Z'$  is the axis of rotation of the scanner,  $Y' = Y$  and therefore  $X'$  is  $X$  rotated about  $Y$  through the angle  $\theta_s$ . Now  $x_m$  and  $y_m$  are the  $x$  and  $y$  coordinates of  $N$ , while  $z_m$  is the height corresponding to the length of  $E_\omega R_\omega$ , which we shall call  $h$ . So, taking  $z_0$  as the distance of the scan base below  $N$ ,  $z_m = z_0 - h$ , so  $h = z_0 - z_m$ .

Now, referring to Figures 1 and 2(b), we define some further points and angles:  $Q_\omega$  is the base of the perpendicular from  $E_\omega$  to the  $X'Y'$  plane;  $S_\omega$  is the point where the beam hits the  $X'Y'$  plane;  $\psi$  is the angle of  $Q_\omega S_\omega$  relative to the radius  $OQ_\omega$ . We take  $Q_0$  to be the position of  $Q_\omega$  when  $\omega$  is 0,  $Q_1$  to be the position of  $Q_\omega$  for which it lies on the  $X'$  axis and  $\varphi_s$  to be the angle between  $OQ_0$  and  $OQ_1$ .

The conversion between the second and third coordinate systems (rotation about the  $Y$  axis) are given by:

$$(X', Y', Z') \rightarrow (X' \cos \theta_s + Z' \sin \theta_s, Y', Z' \cos \theta_s - X' \sin \theta_s) \text{ in } X, Y, Z \text{ coordinates.}$$

$$(X, Y, Z) \rightarrow (X \cos \theta_s - Z \sin \theta_s, Y, Z \cos \theta_s + X \sin \theta_s) \text{ in } X', Y', Z' \text{ coordinates.}$$

The angles in the  $XY$  plane corresponding to  $\varphi_s$  and  $\omega$  are  $\varphi_s^*$  and  $\omega^*$ , respectively, as shown in Figure 2(c). They can be calculated by:

$$\varphi_s^* = \tan^{-1}(\cos \theta_s \tan \varphi_s) \text{ and } \omega^* = \varphi_s^* + \tan^{-1}[\cos \theta_s \tan(\omega - \varphi_s)]. \quad (1)$$

Let the length of  $NO$  be  $d$ , then the height of  $N$  above the  $XY$  plane is  $d \cos \theta_s$ , so we have  $z^* = z_0 - d \cos \theta_s$ . (2)

We will be able to find  $z^*$  when  $d$  is known. We also need  $d$  to find  $x^*$  and  $y^*$ , so we proceed as follows. Let  $r$  be the length of  $Q_\omega S_\omega$ , then we define:

$$C = \rho \cos(\omega - \varphi_s); S = \rho \sin(\omega - \varphi_s); \quad (3)$$

$$C' = r \cos(\omega - \varphi_s + \psi); S' = r \sin(\omega - \varphi_s + \psi); \quad (4)$$

$$\text{For ease of derivation we also define } C^* = \cos(\omega - \varphi_s + \psi) \quad (5)$$

So that relative to the  $X'Y'Z'$  axes, from Figure 2(b)

$$S_\omega = (C + C', S + S', 0) \text{ and } E_\omega = (C, S, d). \quad (6)$$

Now, from Figure 2(d) to find  $R_\omega$  we need the intersection of the line  $Q_\omega S_\omega$  with the  $XY$  plane. The plane has equation  $Z = 0$ , i.e.  $Z' \cos \theta_s - X' \sin \theta_s = 0$  (7)

and the line  $Q_\omega S_\omega$  has general point given by:

$$X' = C + \lambda(C + C' - C) = C + \lambda C', \quad (8)$$

$$Y' = S + \lambda(S + S' - S) = S + \lambda S', \quad (9)$$

$$Z' = d + \lambda(-d) = d(1 - \lambda), \quad (10)$$

where  $\lambda$  can take any value. We want to find the value of  $\lambda$  corresponding to  $R_\omega$ .

Substituting into equation (7) and simplifying, we obtain:

$$\lambda(C' \sin \theta_s + d \cos \theta_s) = d \cos \theta_s - C \sin \theta_s. \quad (11)$$

We also need to find  $d$  and  $\lambda$  in relation to  $h$ :

$$\text{From equations (8)-(10), } h = \text{length of } E_\omega R_\omega = |(\lambda C', \lambda S', -\lambda d)|, \quad (12)$$

$$\text{So } h^2 = \lambda^2 [(C')^2 + (S')^2 + d^2] = \lambda^2 (r^2 + d^2)$$

$$\text{But, from Figure 2(d), } r = d \tan \theta, \quad (13)$$

$$\text{so } h^2 = \lambda^2 d^2 (1 + \tan^2 \theta) = \frac{\lambda^2 d^2}{\cos^2 \theta}.$$

Therefore, taking the positive root, because  $R_\omega$  is below  $E_\omega$ , we have

$$h = \frac{\lambda d}{\cos \theta} \text{ and } d = \frac{h \cos \theta}{\lambda}. \quad (14)$$

$$\text{We also obtain } C' = rC^* = C^* d \tan \theta = \frac{C^* h \sin \theta}{\lambda}.$$

Now we can solve for  $\lambda$  by substituting for  $C'$  and  $d$  in equation (10) to give:

$$hC^* \sin \theta \sin \theta_s + h \cos \theta \cos \theta_s + C \sin \theta_s = \frac{h \cos \theta \cos \theta_s}{\lambda},$$

$$\text{so } \lambda = \frac{h \cos \theta \cos \theta_s}{(hC^* \sin \theta \sin \theta_s + h \cos \theta \cos \theta_s + C \sin \theta_s)},$$

$$\text{and } d = \frac{h \cos \theta}{\lambda} = h \cos \theta + \tan \theta_s (hC^* \sin \theta + C). \quad (15)$$

Now  $X' = C + \lambda C'$  and  $Y' = S + \lambda S'$  from equations (8) and (9), so in  $X$  and  $Y$  coordinates we have, relative to O:

$$X_0 = (C + \lambda C') \cos \theta_s + d(1 - \lambda) \sin \theta_s,$$

$$Y_0 = S + \lambda S'.$$

To obtain the  $XY$  coordinates relative to N, we have to subtract  $d \sin \theta_s$  from  $X_0$ , giving

$$X = (C + \lambda C') \cos \theta_s - \lambda d \sin \theta_s, \quad (16)$$

$$Y = S + \lambda S'. \quad (17)$$

So, finally from Figure 2(c), rotating  $X$  and  $Y$  by  $-\varphi_s^*$ ,  $x^*$  and  $y^*$  are given by:

$$x^* = x_m + X \cos \varphi_s^* - Y \sin \varphi_s^*,$$

$$y^* = y_m + X \sin \varphi_s^* + Y \cos \varphi_s^*.$$



## **Appendix B**

This appendix contains a book chapter submitted for approval to Dipak Laha, Jadavpur University, India and Purnendu Mandal, Lamar University, USA for inclusion in their publication 'Handbook of Computational Intelligence in Manufacturing and Production Management'. The chapter presents a synopsis of the work in this thesis.

# INTELLIGENT LASER SCANNING OF 3D SURFACES USING OPTICAL CAMERA DATA

A. Denby, J.F. Poliakoff, C. Langensiepen, N. Sherkat, The Nottingham Trent University, UK.  
Email: janet.poliakoff@ntu.ac.uk

**Key words:** Image processing, laser scanning, CAD/CAM, CCD camera, sensor fusion

## Abstract

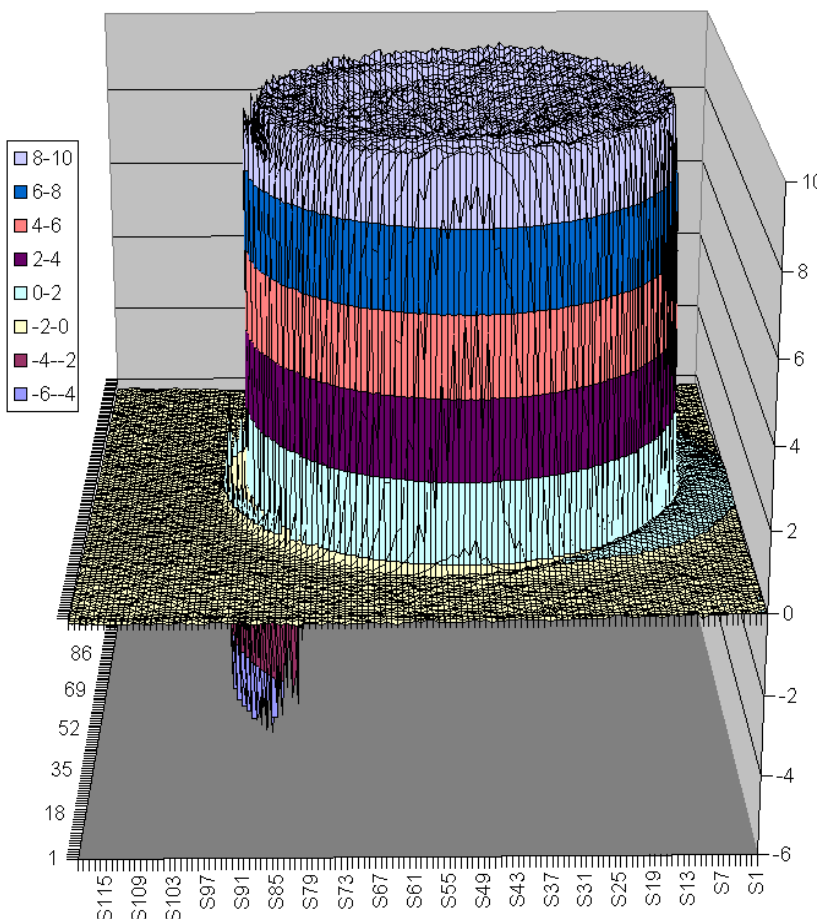
In CAD/CAM, reverse engineering involves obtaining a CAD model from an object that already exists. An exact replica can then be produced, or modifications can be made before manufacture. Single-perspective triangulation sensors provide an inexpensive method for data acquisition. However, such sensors are subject to localised distortions caused by secondary reflections or occlusion of the returning beam, depending on the orientation of the sensor relative to the object. This chapter describes an investigation into integrating optical camera data to improve the scanning process and reduce such effects, and intelligent algorithms, based on image analysis, which identify the problem regions, so that the sensor path and orientation can be planned before the scan, thereby reducing distortions.

## 1. Introduction

Ideally an object is designed on a CAD system to provide the data needed to control the CAM equipment to manufacture the object. However, there is often a need to copy objects for which no prior CAD data are available, e.g. when making replacement parts. Machining such objects by hand is

possible but expensive, as is redesigning the objects on a CAD system. Therefore there is a real need for an inexpensive method for generating the required data from the object which maintains an acceptable degree of accuracy.

One approach is to use a laser sensor to measure the surface. Unfortunately, laser scanning is subject to localised distortions, which are often caused by occlusion or secondary reflections of the beam, depending on the orientation of the laser head relative to the object. Without prior knowledge of the object, a 'blind' scan must be implemented. We have investigated the integration of an optical camera into the system to provide such knowledge. Image analysis allows the path and orientation of the laser sensor to be planned before the scan, thereby reducing the distortions. Scanning time can also be shortened by reducing scan resolution in 'low interest' regions.



*Figure 1: A single scan of a small bottle top with orientation parallel to the x-axis (i.e. left to right). The vertical scale is exaggerated to help show the errors, which can be seen where the edge is roughly perpendicular to the sensor orientation: upward and downward 'spikes' on the left; smaller 'bow wave' errors on the right, which extend further from the object.*

It has been found that simple edge detection algorithms such as Canny can determine a single best orientation, but a combination of algorithms is needed to eliminate noise and create continuous edge segments, which can then be used to develop scan regions of appropriate orientation. We have developed new vectorisation algorithms to identify edge segments. Calibration of the camera image relative to the scanner is important to avoid errors. Discrepancies between scan data from different orientations can be prevented by careful calibration of the scanner rotation system.

## 2. Background

In traditional ‘forward’ engineering, concepts and models are transformed into real parts. Reverse engineering starts with real parts or prototypes and transforms them into engineering models. Typically, the process begins by measuring an existing object to provide a model, in order to exploit the advantages of CAD/CAM technologies. Such techniques are used in a wide variety of applications, including medicine and animation as well as more traditional production. A typical application is the re-engineering of an existing structure for input into a CAD or other 3D modeling program, where analysis and modifications are required to make a new, improved product. The data acquisition phase is a crucial step in this procedure and data acquisition methods can be either tactile or non-tactile.

Laser triangulation is a popular non-tactile data acquisition method in which a laser beam is projected onto the surface of interest and the reflected spot is detected by one or more photosensitive devices. The position of a surface point is then calculated using triangulation. Laser triangulation can acquire data at very fast rates; however the technique is subject to errors, as shown in Figure 1. Before describing the types of errors that occur, we explain how the triangulation process works.

The laser scanner consists of a unit with an emitter and detector which moves over an object and outputs readings corresponding to the distance of the object from the scanner. The emitter projects a laser beam onto the object in a (normally) vertical direction, as shown in Figure 2. For ease of explanation we have based our descriptions on the assumption that the beam is vertical but the principle can equally well be used for other configurations. The sensor detects light returning from the spot where the beam hit the object and measures the direction at which it returns, here indicated by angle  $\phi$ . The distance of the spot from the emitter can then be calculated by  $h = w \cot \phi$ , where  $w$  is the distance between emitter and sensor. As the scanner moves over the object, the positions of points on the surface are measured and collected to form the point cloud.

Our group has worked for many years at the interface of engineering and computer science. We have made contributions in the fields of control of CNC machines (Chow, Poliakoff & Thomas, 2002, Poliakoff, Chow, Orton, Howson & Al-Dabass, 2005), measurement of surfaces (Wong, Poliakoff &

Thomas, 2001, Sacchi, Poliakoff, Thomas & Häfele, 2004, Denby, Langensiepen, Poliakoff & Sherkat, 2005). The work described in this Chapter arose from an investigation into the errors that occur in laser triangulation. Wong (2002) found that the majority of these errors fall into three broad categories: (systemic) noise, transitional errors caused by changes in reflectivity across the object and errors due to the geometry of the object. Wong investigated how to reduce the geometric errors in the third category, which are caused when the object itself interferes with the measurement of the height of the primary spot. He found that sometimes there are secondary reflections of the light from the primary

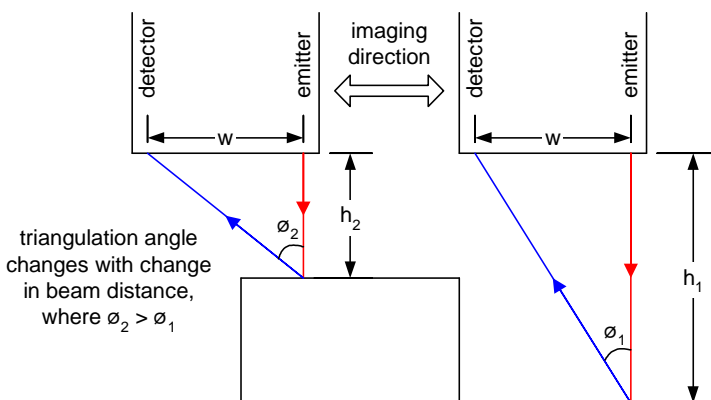


Figure 2: Diagram to illustrate how the heights of different parts of the object are measured using triangulation. The triangulation angle,  $\phi$ , varies as the height changes; the height is calculated as  $h = w \cot \phi$ , where  $w$  is the distance between emitter and detector.

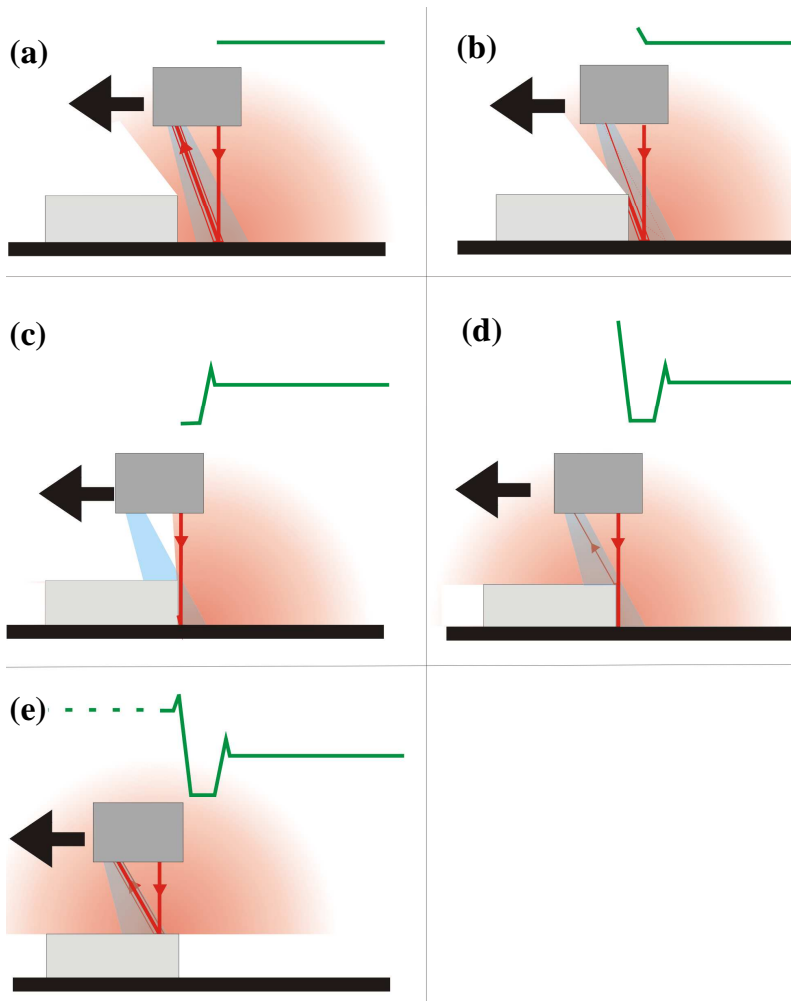


Figure 3: Illustration of how the ‘spikes’ distortion occurs for a simple object. The scanner is shown moving from right to left and building up the output profile in green, although the errors obtained do not depend on the direction of motion. At (a) and (e) there are no distortions. Occlusion begins at (b) with part of the beam occluded and a small rise in the output. At (c) the beam is completely occluded and the output is zero, giving a trough, whereas at (d) part of the spot is on the upper surface and another small rise is seen in the output. If the surface has high reflectivity, light from secondary reflections during complete occlusion (c) may produce the large rise or ‘spike’ effect.

Initially a number of complete scans of the object are made with different orientations. Intelligent software, based on comparisons between the scans, is used to identify problem regions and associate an edge with each such region. Then, the appropriate scan is selected with orientation chosen to minimise the error on that region. Wong’s method has resulted in much improved scan quality. Unfortunately, the major drawback of this approach is a greatly increased overall scan time, because at least eight complete scans (at 45° intervals) are needed. Typically a single scan of 0.05 mm pitch requires about 1 hour to execute for an object of 30 sq cm., so 8 complete scans require a about 8 hours. Our latest investigation has aimed for similar improvement but without the need for repeated complete scans.

### 3. Optical Camera Data Integration

Most of the data collected using Wong’s approach is eventually discarded, having only been used in order to identify error regions and select the appropriate scan orientation for each such region. Our new approach is to recognise the problem regions *prior* to the laser scan process, in order to determine

spot onto other parts of the object within the sensor’s field of view; if these are also detected by the sensor, there will be an error in the reading obtained. In other cases the primary spot could be occluded from the sensor by another part of the object, and then the error is likely to be even larger.

Figures 3 and 4 summarise Wong’s findings. When the reflected signal is occluded from the detector by part of the object, e.g. at (b), (c) and (d) in Figure 3, false readings with substantial errors are obtained (which we refer to as ‘spikes’). In other cases, e.g. at (a) and (b) in Figure 4, secondary reflections also cause false readings, for which the errors are not so large but more extensive (which we refer to as ‘bow wave’), as also shown in Figure 1. These ‘geometric’ errors close to edges of the object are worst when the laser scan head is oriented approximately *perpendicular* to the edge (to the left and right in Figure 1). Whereas, if the laser is oriented *parallel* to the edge, the errors are minimised. Figure 5 shows how the errors increase as the relative orientation changes from parallel towards perpendicular.

Wong addressed these ‘geometric’ errors by exploiting the fact that knowledge of the orientation of an edge will allow the system to identify the laser unit orientation likely to produce least distortion near that edge.

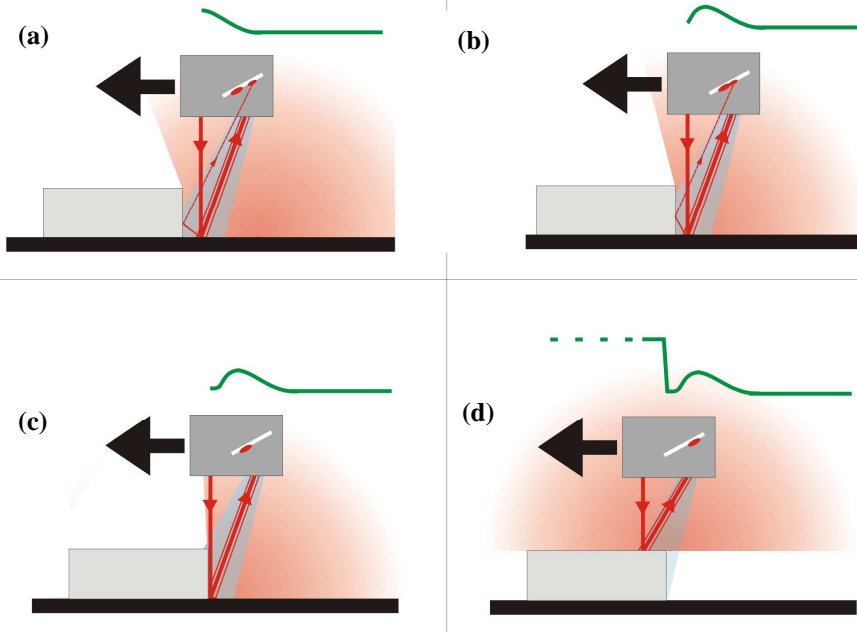


Figure 4: Illustration of how the 'bow wave' distortion occurs for a simple object. The scanner moves from right to left, as in Figure 3. The bow wave grows as it approaches the object as secondary reflections cause the output to rise until a maximum bow wave is reached at (a). At (b) it is still closer to the object, so that the secondary reflections have a smaller effect. Just before the spot reaches the edge (c) the output dips further. When the spot is on the top of the object at (d) there are no distortions.

an optimal scanning strategy for the object. By integrating an optical camera with the laser triangulation system we can obtain knowledge of the object's geometry prior to scanning. Then we can identify regions where optimal choice of sensor orientation is crucial, rather than relying on post-processing of multiple scans to do this. The path planning then involves selecting the best sensor orientation for different regions of the object, thereby minimising such 'geometric' errors. However, unlike Wong's method, much redundancy in scanning the same region many times is avoided, thus reducing the total scan time.

We use information captured from a simple (2D) CCD camera as an *a priori* guide to orientation and path-planning

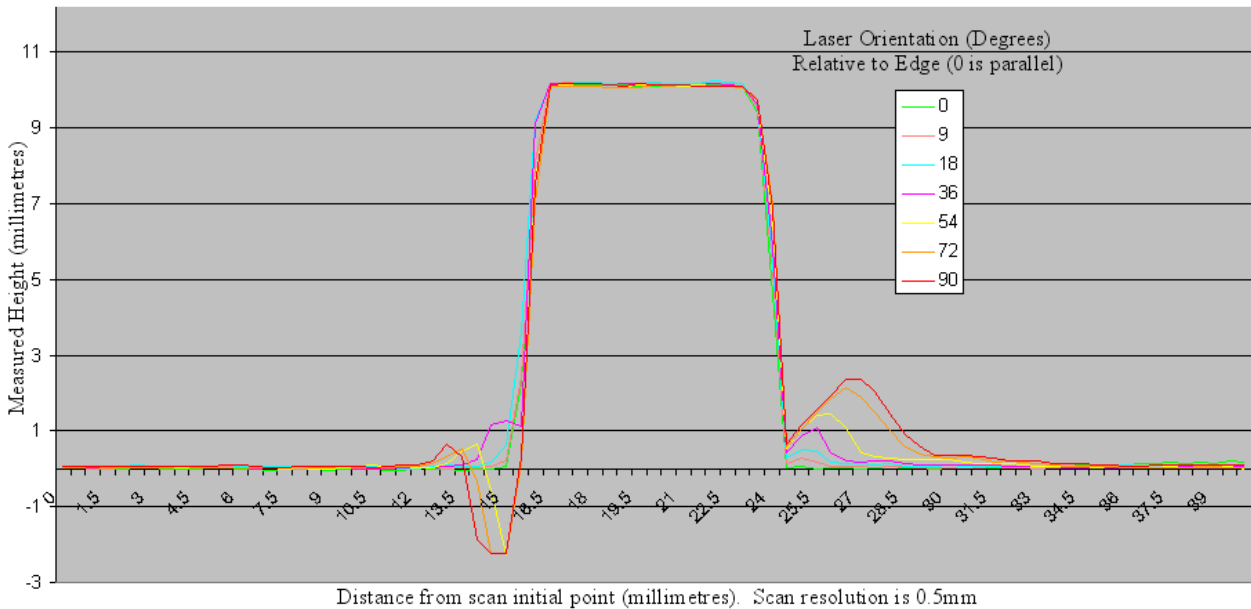


Figure 5: A plot of the scan output for different orientations of the scanner, showing how the distortion are least at 0°, when the scanner is oriented parallel to the edge, and largest at 90°, when it is perpendicular. The scale in the vertical axis is exaggerated.

for the scanning laser. The camera image is analysed using edge detection techniques and then vectorised to provide an edge map. A scanning path plan that minimises errors due to object geometry is generated based on the edge map and corresponding region-segmented images.

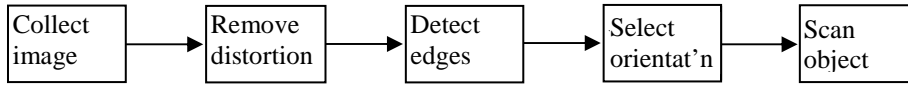


Figure 6: Overview of Method 1

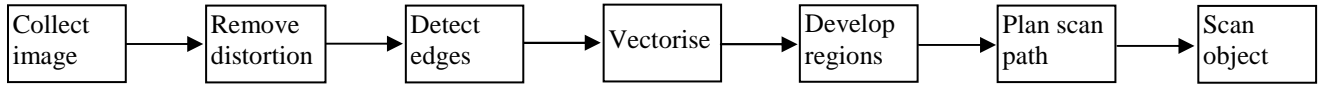


Figure 7. Overview of the Methods 2 and 3

The simplest strategy, Method 1, involves performing a single scan with a fixed orientation, as illustrated in Figure 6. The (small) improvement here is that the orientation for the scan is chosen to minimise the number of error points in the point cloud. This method is not much slower than a standard single scan, because the only additional time is for the edge detection software to run. However, it has the disadvantage that generally there will still be many distortions in the resulting point cloud, so we include it merely for comparison.

A better strategy, Method 2 (Figure 7), is to generate a number of different partial scan paths, each covering a number of regions in the image that share a common ‘optimal’ scan orientation. Figure 8(a) – (e) illustrates the idea behind this method for a simple L-shaped object. A number of partial scans are then executed, with minimal overlap between them, with the scan head being re-oriented between the scans. Remaining parts can be scanned with any orientation. This method is slower than a single scan but faster than Wong’s method. Where the regions overlap some procedure is needed to resolve any conflict between the data from the different regions, for example by simple averaging or by some sort of blending. If the regions can be chosen without any overlaps, then this method can be replaced by the approach of Method 3.

As illustrated in Figures 7 and 8(f), Method 3 uses a single scan sweep over the object with continuous real-time automatic optimisation of the laser scan head orientation. This method requires careful monitoring to ensure continued accuracy throughout the scan and requires equipment that is capable of controlling the rotation of the scan head as needed in real time. Again this method is faster than Wong’s method but slower than a single scan. It may be faster than Method 2 but that could depend on the complexity of the object’s geometry. The time taken may have to be slower than a single scan with fixed orientation, in order to accommodate the real-time rotation of the scan head.

#### 4. Our Investigation

In order to investigate the three methods described above, we chose a slightly different approach. Because the scanning process is very time-consuming, we wanted to avoid performing many experiments with different path plans. Therefore we came up with the strategy of virtual scans based on a number of complete scans covering all possible scan orientations within a certain approximation. Then a path plan could be ‘tested’ using a virtual scan, by selecting the required values from the scan data for the relevant orientation. Further work would be needed for Method 3 but many questions can be answered by using such a virtual scan.

So, for each object a set of 10 complete scans was obtained at  $18^\circ$  intervals, thus covering a range of  $180^\circ$ . This then provided a scan of any orientation to within  $9^\circ$  of the required orientation, because two orientations  $180^\circ$  apart can be covered by the same scan. Again for further refinements of the proposed methods some parts of the other 10 scans might be needed.

In this way we could also perform a virtual partial scan based on the scan data already collected. Each virtual partial scan is performed by choosing the appropriate data from the pre-existing scan. Although the initial collection of complete scan data for all the orientations was time-consuming, it avoided much repeated performance of partial scans during the investigation. This approach does not allow us to choose every possible orientation but we have found that it is sufficient to demonstrate the principle. Figure 9 is a modification of Figure 7 showing how the virtual scan is incorporated.



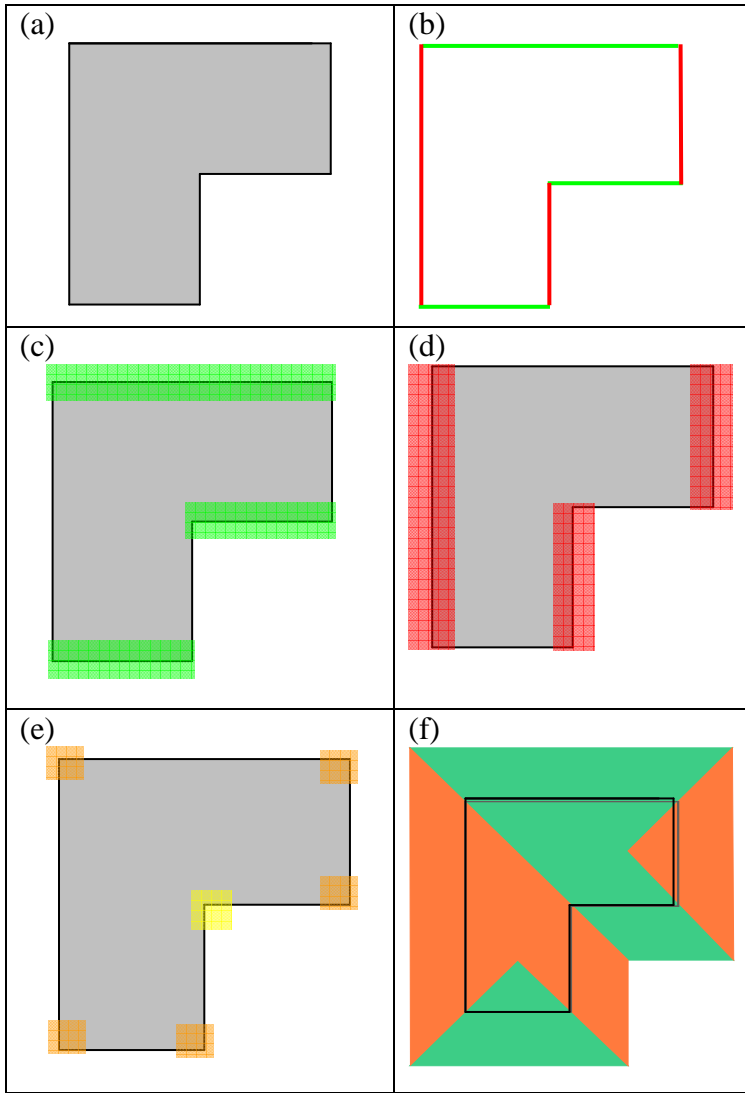


Figure 8: Illustration of the idea behind Methods 2 and 3 for a simple L-shaped object (a). The detected vectors are shown in (b), with red representing horizontal ( $0^\circ$ ) and green vertical ( $90^\circ$ ). Here Method 2 uses scan region Algorithm A and the scan regions for  $0^\circ$  and  $90^\circ$  are shown in (c) (green) and (d) (red), respectively. The places where the scan regions overlap are shown in (e), where the orange ones are expected to give good results, because there is an outer corner. In the case of an inner corner, shown in yellow, it may be difficult to resolve the output value. Method 3 uses scan region Algorithm B and there are no overlaps (f); the red region is to be scanned at  $0^\circ$  and the green scanned at  $90^\circ$ . (The grey region can be scanned with any orientation.)

Section 5 describes the image processing and edge detection algorithms needed to produce a map of all the edge pixels from the image. From these the optimum orientation can be found for Method 1. The vectorisation and region development algorithms are presented in Section 6, which can then be used to produce the path plans needed for Methods 2 and 3.

## 5. Image Processing for Edge Detection

Since colour images provide more information than grey value images, more detailed edge information might be expected from color edge detection, however Novak & Shafer (1987) found that 90% of edges are approximately the same in grey value and in color images, although it is possible that the remaining 10% may make a significant difference to the overall edge continuity. Although colour image output was available from the digital cameras, it was decided initially to implement several common edge detection methods using greyscale images to test the hypotheses presented in this project because of the complexity of colour edge detection methods. For this project it was often necessary to spray-coat the objects being scanned and imaged in order to reduce the chance of specular reflections occurring in the scan data. The sprayed surface means that images are generally monochromatic so there is no advantage (at this stage) in using colour images.

The image processing begins with compensation for distortions in the image caused by deficiencies in the camera. Before edge detection is performed it may be useful to smooth the image. For both smoothing and

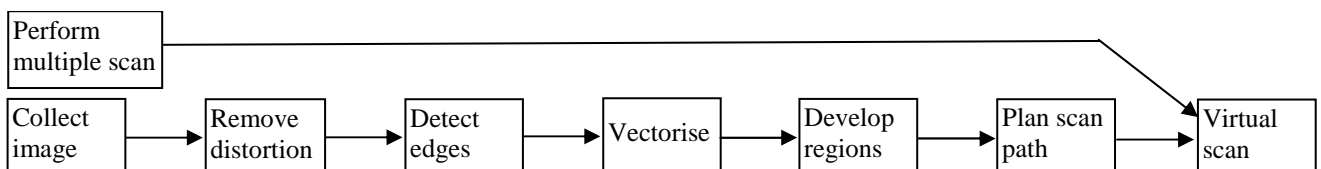
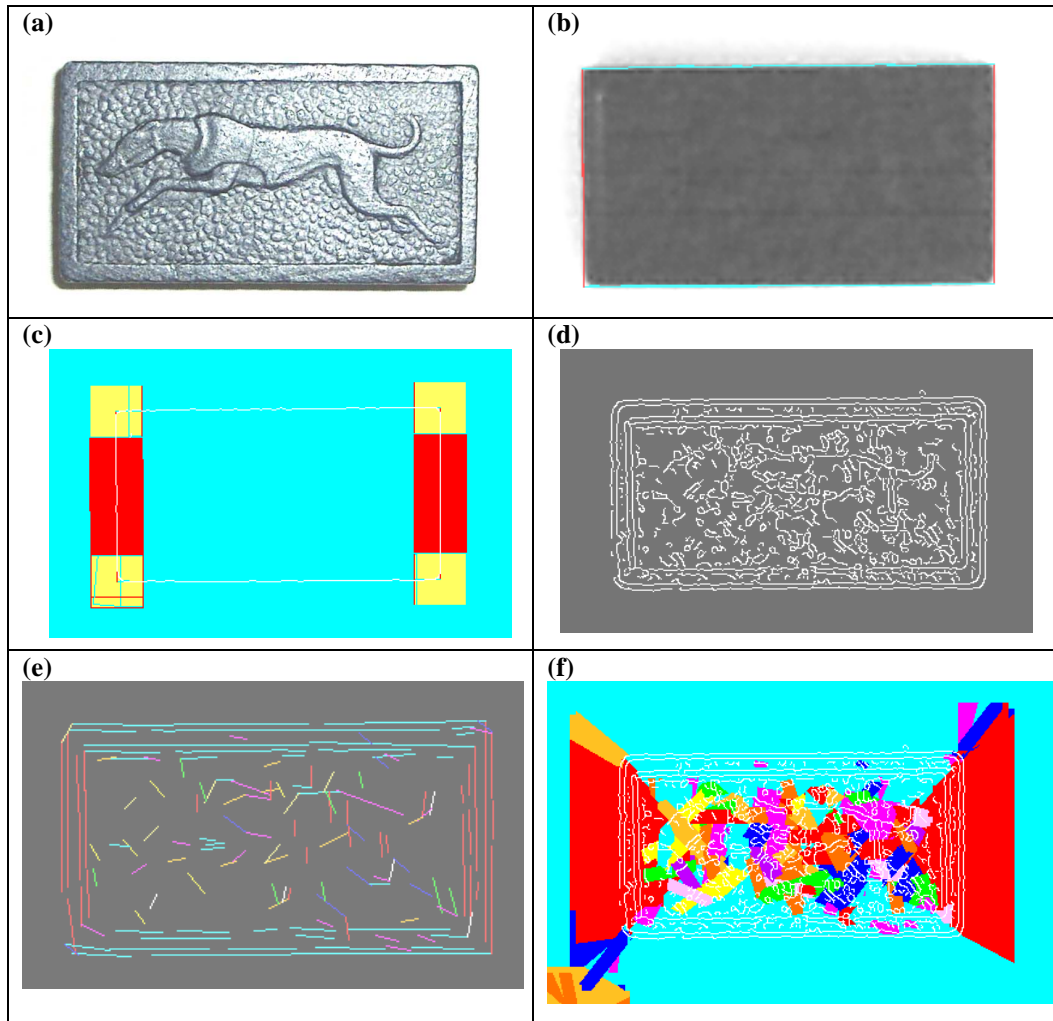


Figure 9. Overview of the scan process used for our investigation.



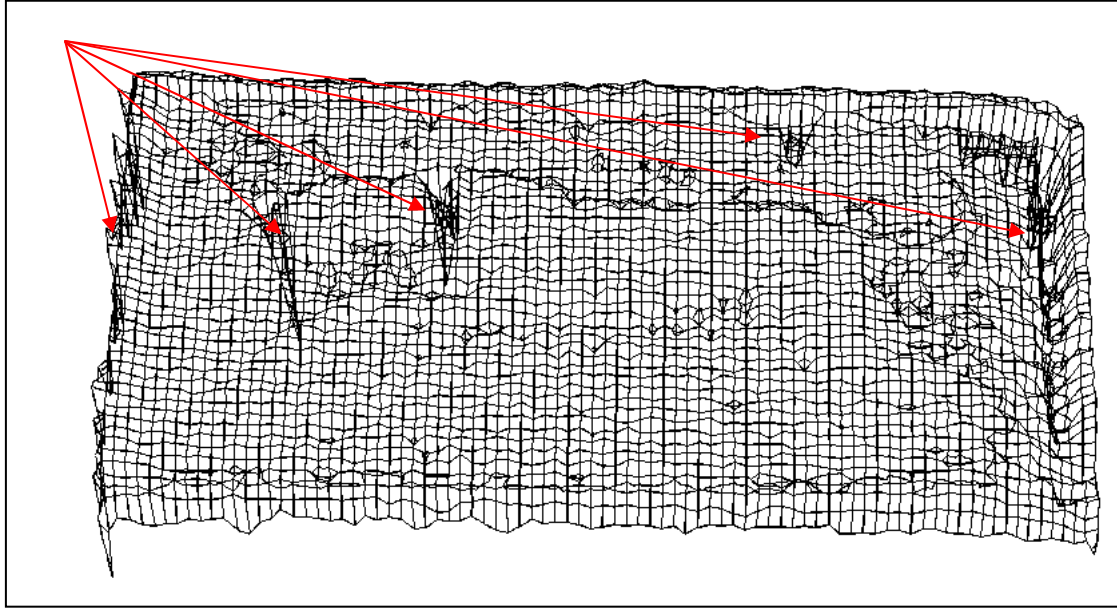
*Figure 10: Image processing results for a black domino, initially unsprayed: (a) photograph of the domino; (b) vectors; (c) scan regions using Algorithm A with detected edges superimposed in white. After the domino has been sprayed, more detail is detected: (d) edges; (e) vectors; (f) scan regions using Algorithm B (where the regions at bottom left were caused by an alignment spot). The different colours indicate different orientations, e.g. turquoise for  $0^\circ$ .*

edge detection the method of ‘convolution’ is often used. At each position a mask is used to calculate a weighted sum of nearby pixels to replace the original pixel value. For smoothing, the mean filter takes a simple average, while the Gaussian mask is symmetrical with the highest weight in the centre.

Edge detection involves the identification of pixels on the image where there are discontinuities or abrupt changes in grey level (intensity) or colour, or the gradient of this intensity or colour. Edges usually correspond to significant variation in reflectance, illumination, orientation and depth of surfaces and are typically associated with photometric, geometric and physical characteristics of objects within the image (Ziou, 1998). A wide range of methods have been used for edge detection, some of which are described here. First and second derivative operators have been used to identify edge pixels, which then require further processing to thin the detected edges. Other approaches include region growing, which can identify regions which are not edges, and segmentation using texture, as explained below.

Examples of first derivative operators are the Roberts Cross operator (Ziou, 1998), the Sobel operator (Lyers, 1988), the Prewitt gradient method (Prewitt, 1970) and the Frei-Chen method (Ziou, 1998). These all involve a different square masks for a ‘convolution’ with the image to produce a new output image (Low, 1991). The Canny edge detector (Canny, 1986) employs a Gaussian smoothing function and simple first derivative masks for detection. The technique was extended by Deriche (1987) to





*Figure 11: The combined scan results for the domino using the plan in Figure 10(c), showing downward spike errors (indicated by the red arrows) in the top surface of the domino. The scale in the vertical axis is exaggerated to help show errors.*

employ both a gradient magnitude and a gradient orientation map to determine the direction for the edge-tracking and to assign an orientation to each edge pixel.

In conjunction with a thresholding method, such as the edge-following technique used by the Canny operator, most of these edge detectors provide a useful edge response for vectorisation. However the Frei-Chen masks do not provide a means of recovering orientation information. A selection of established edge detection algorithms have been implemented using a common code platform in order to compare them in a controlled manner and provide unbiased results. The Roberts Cross, Prewitt gradient, Sobel, and Frei-Chen methods were implemented and compared with the Canny edge detector without the Gaussian smoothing and the results are shown in Table 1.

Operator	Mask Size	Number of Masks	Mean time to execute (ms) (averaged over 5 samples)
Robert's Cross	2x2	2	245.3
Prewitt Gradient	3x3	2	325.5
Sobel	3x3	2	320.6
Frei-Chen	3x3	9	850.5
Canny (convolution only)	1x3	2	233.5

**Table 1. Comparison of time to execute first derivative operators (640x480 greyscale image)**

The Sobel operator also generally produces considerably higher output values for similar edges, compared with the Roberts Cross. The Roberts and Sobel gradient masks are more sensitive to diagonal edges. The Prewitt gradient mask is more sensitive to horizontal and vertical edges. The Frei-Chen edge detector has equal sensitivity for diagonal, vertical and horizontal edges. However, the time taken is more than three times longer than for the Canny edge detector, which we therefore used for our investigation.

A number of edge detection algorithms have been evaluated for their suitability for the identification of edges in physical objects for path planning in preparation for laser scanning. Our findings support the

view of Heath et al. (1998) that the Canny operator provides the best general edge detector where the parameters can be tuned for each image.

The Canny algorithm uses Gaussian smoothing followed by computation of the gradient magnitude and direction, giving a pair of values for each pixel. The edges are then localised by using a process of ‘non-maximal suppression’, which provides an improvement to the basic skeletonisation technique. The idea is to determine the local maximum of the gradient magnitude and then track along the top of the gradient ‘ridge’ in both directions along the orientation of the expected edge (i.e. perpendicular to the gradient direction). A pixel is considered to be the local maximum in this context if the magnitude of the gradient for that pixel is greater than that of the two neighbouring pixels in the direction of the gradient. All edge pixels that are not maximal (i.e. on the top of the edge) are set to zero, giving a line of single pixel width in the output.

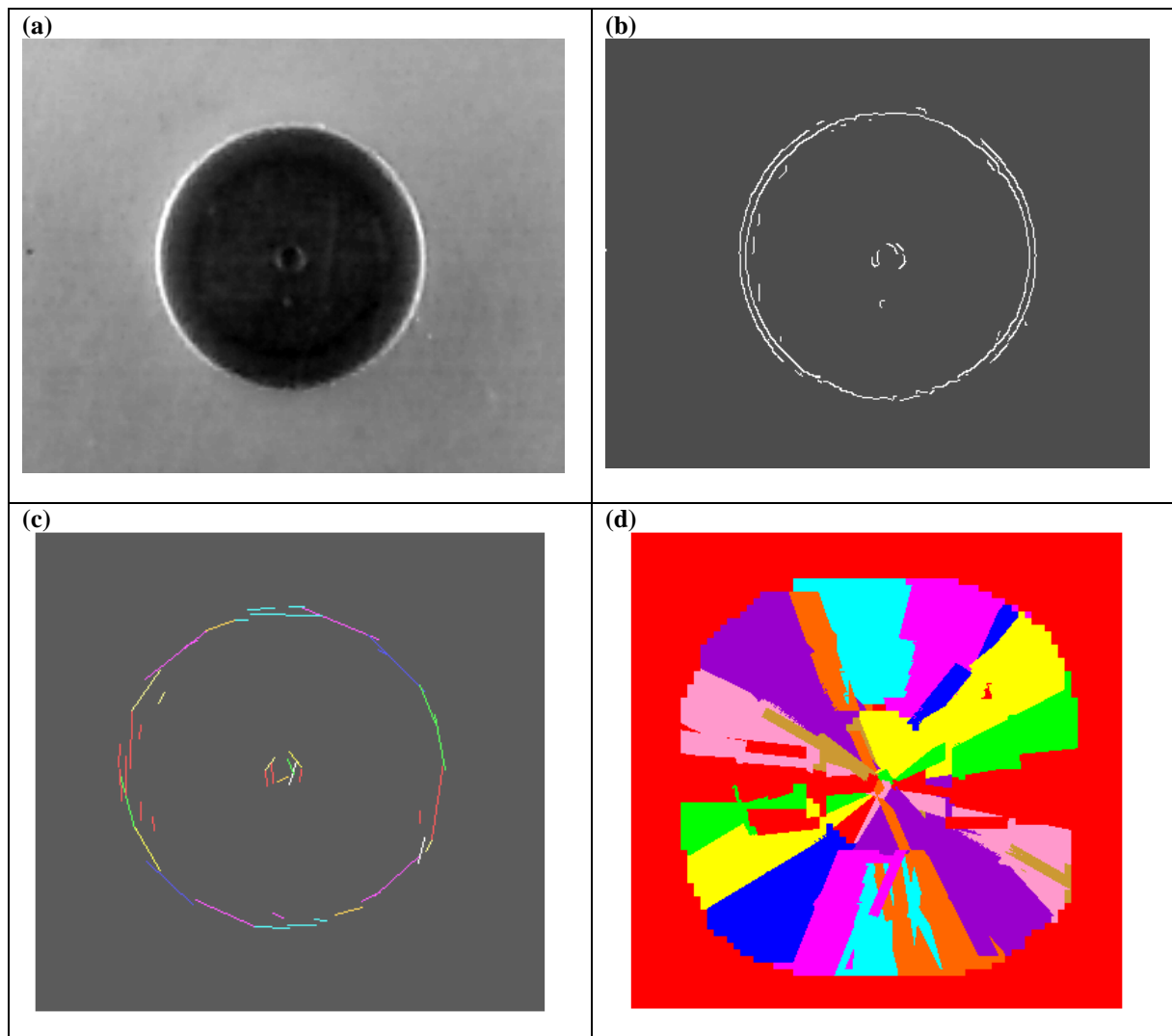
Thresholding techniques are reviewed by Sezgin & Sankur (2004) and we use thresholding as a method to localise the edges. The tracking process is controlled by two thresholds:  $T_{max}$  and  $T_{min}$  where  $T_{max} > T_{min}$ . Edge tracking can only begin at a point on a locally maximal edge pixel with a gradient modulus higher than  $T_{max}$ . Tracking then continues in both directions out from that point until the height of the ridge falls below  $T_{min}$ . This edge hysteresis helps to ensure that noisy edges are not broken up into multiple edge fragments. Usually, the upper tracking threshold can be set quite high, and the lower threshold quite low for good results. Setting the lower threshold too high will cause noisy edges to break up. Setting the upper threshold too low increases the number of spurious and undesirable edge fragments appearing in the output.

Texture information is an important consideration for edge detection but we have not used it here. Pure texture segmentation gives only a coarse segmentation, so it can only be used as an auxiliary tool to check segmentation and texture parameters for the segmented regions. Haralick et al.(1973) provide definitions of texture and derive a number of texture parameters including contrast, correlation, direction, entropy, homogeneity and uniformity. One approach commonly used to handle texture is to smooth the image using a Gaussian or other blurring filter. However, this can cause problems, because, as the strength of the blurring filter increases, it becomes more difficult to detect the position of the edges accurately (i.e. edge localisation suffers) and also fine detail which we may want to keep becomes lost along with the texture ‘noise’.

Corner finding has also been investigated in the context of edge detection. Kitchen and Rosenfeld (1982) used a local quadratic fit to find corners. Wang and Binford (1994) modelled the effects of shading on the direction of the image gradient, creating a detector insensitive to shading. In practice most corner detectors are usually not very robust and often require expert supervision to prevent the effect of individual errors from dominating the recognition task. Smith and Brady (1997) proposed the Smallest Univalued Segment Assimilating Nucleus (SUSAN) corner detector, which performs well in inherently noisy ‘real world’ images because it does not rely on image derivatives. It may be that corner finding could be used to identify potential problems where two or more edges meet but we have not used it so far.

Split-and-merge edge detection is a two stage process that combines the advantages of both region growing and region splitting methods. Firstly the image is split recursively until each region in the image meets the specified criteria of homogeneity then adjacent regions are merged together if they satisfy the same criteria. Because both split and merge processing options are available, the starting segmentation does not have to satisfy any of the homogeneity conditions. In more complex images even more complicated criteria may not be enough to give acceptable results. The split-and-merge techniques introduced by Chen and Pavlidis (1981) and later developed by Spann & Wilson (1985) use a linked pyramid and statistical decision criteria to combine global and local region information.

The white lines in Figures 10(c), 10(d) and 12(b) show the results of edge detection for the domino shown in Figure 10(a), unsprayed and sprayed, respectively, and the bottle top shown in Figure 12(a), sprayed. The next stage is to vectorise the identified edge pixels, so that they can be used as the basis for developing scan regions, as described in the next section.



*Figure 12: Image processing results for the small bottle top from Figure 1, which was also sprayed: (a) photograph of the object; (b) edge detection; (c) vectorisation and (d) scan regions obtained using Algorithm B. Again, the different colours indicate different orientations, e.g. turquoise for  $0^\circ$ .*

## 6. Vectorisation and Scan Region Development

Many vectorisation methods have been proposed in the scenario of converting line drawing images (e.g. engineering drawings) directly into CAD models, which is in many ways analogous to the process that is being investigated in this project. However none of the methods proposed work perfectly (Tombre, 1998). The output of vectorisation should represent the shape of original image as faithfully as possible but, the vectors do not always correspond with ‘ground-truth’ graphic objects (Tombre, 2000). Some post-processing is usually necessary to rebuild graphic objects from vectors with geometric constraints. Many of these methods are based on skeletonisation or other thinning methods that are not required here (since the edges are already thinned to a single pixel width by the non-maximal suppression process). One method of storing the component pixels in vectorisation is chain encoding, such as Freeman Chain Codes (Freeman, 1970) in which the direction of neighbouring edge pixels are stored sequentially according to their relative positions. Line and arc-fitting algorithms are often employed to convert the original image into a low-level vector format represented by short line segments and short arcs. Line-fitting methods are popular but since they depend on approximating a sequence of adjacent line segments, a ground-truth line cannot be recovered correctly if some parts of it are missing or have serious distortions (Hori, 1993).

The Hough transform is one method that has often been used to detect known geometrical shapes such as lines and circles (or other known shapes) in images (Ballard, 1981). The main advantage is that it is tolerant of gaps in feature boundaries and is relatively unaffected by image noise. However it is computationally- and memory-intensive (Risse, 1989). A number of authors have proposed improvements to the standard Hough transform to improve line detection and localization. Ji & Xie (2003) review the approaches taken by a number of authors contributing to the use of Hough Transforms, as well as proposing a method by which edge localisation may be improved. Conceptually, the Hough transform considers lines at all possible positions and orientations and counts the number of pixels that fall on each line. A ‘transform space’ is created where points in the ‘Hough space’ map to lines in the image space. This method finds many lines in the image, but we have found that it has several unwanted effects. Firstly, quantization of the pixels in the image space and of the accumulator cells in Hough space can lead to a cluster of points in Hough space. When these points are mapped back into the image space they form a group of lines of slightly different orientations intersecting at a common point, giving a ‘bow tie’ effect. Secondly, lines that pass through many pixels are favoured by the accumulator because there is more pixel evidence for them, which means that shorter lines will be removed by thresholding. Such short lines are equally valid as edges which cause the distortions with which we are concerned. We investigated a ‘windowed’ Hough approach to avoid losing shorter lines but found it even more computationally intensive than the standard Hough.

For this application we have developed a vectorisation algorithm which can overcome the problems encountered with the other methods. The idea is to select each edge pixel in the image and attempt to ‘grow’ a straight line starting at that pixel. All eight neighbouring pixels are examined in turn and when an edge pixel is found the ‘growth’ can begin. The line (defined by a start and end pixel) is grown, together with an accompanying list of pixels which have contributed to the development of the line. Growing continues until a stopping condition is reached. The first stage is to seek to extend the line by looking for an edge pixel, firstly at the pixel closest to the extension of the line and adjacent to the end pixel of the line, and then at the pixels on either side (relative to the line). If one or more new edge pixels are found, the one closest to the line is added to the list of pixels and a new line is fitted by least squares fitting (see below for more details). Then the signed distances of all the pixels in the list from the new line are calculated. The growth is stopped if one (or more) of the following cases holds:

1. no extension is possible (no edge pixels found);
2. the new edge pixel is already in the list;
3. the new line has a different start pixel from the original line;
4. the change in direction of the new line compared to the previous one exceeds a given tolerance: the tangent of the tolerance angle is inversely proportional to the length of the line (so the tolerance angle is reduced as the line grows);
5. the distance of one or more of the list of pixels from the new line exceeds a given tolerance;
6. the sum of the signed distances exceeds a given tolerance;
7. the growth becomes one-sided (i.e. there is a sequence of pixels in the list with the following property: the first is at a distance greater in magnitude than the mean distance, all are on one side of the fitted line and the number at greater distance than the previous one exceeds a given tolerance);
8. the line overlaps other lines already discovered, either completely or very closely, so it can be discarded, because nothing new will be found;

The mean distance is calculated as the sum of the signed distances of all the pixels from the fitted line. After the least squares fitting has been done, the start and end pixels of the new line are chosen as follows. If the magnitude of the gradient of the fitted line is less than 1, the y value of the start pixel (respectively end pixel) for the new line is changed, if necessary, so that it lies on the fitted line, and otherwise the x value is changed.

We have found that the above algorithm sometimes produces lines which deviate considerably from the previous trend by the addition of the last few pixels, with the effect of moving the line direction too far

in small increments. This situation is not prevented by conditions 5, 6 and 7, unless the tolerances are made smaller to prevent this. However, with the smaller tolerances many lines are terminated prematurely. Therefore when the growth is stopped, an attempt is made to ‘rewind’ the line until the magnitude of the distance for the last pixel is less than the magnitude of the mean distance for the rewound line. This is done, rather than merely testing for the above condition at each stage of growth, because sometimes the line direction can ‘return’ closer to the trend after a small deviation.

In order to keep the vectors fitted closely to the edges discovered in the image the initial vectorisation tolerance parameters must be quite strict. Simple (artificial) images are reduced to a minimum sufficient set of vectors quite efficiently using this method, however in ‘real’ images there is often some redundancy in vectors, given that the edges are often curved and several straight lines may be fit around that curve. This often leads to the generation of many short line segments, which can sometimes be merged. There are also situations where discovered vectors can be merged, because the initial tolerances caused the break-up of long vectors into shorter segments. The vectorisation algorithm may also result in the generation of parallel overlapping segments from a single line. In such cases it is possible to merge these segments into a single vector using a ‘combine’ algorithm. Figures 10(b), 10(e) and 12(c) show the results of vectorisation for the edges shown in Figures 10(c), 10(d) and 12(b) as white lines. The circular shape of the bottle top consists of short lines at different orientations.

The development of scan regions based on the vectors can be done in various ways. We have implemented two different such algorithms, A and B, the first of which is only appropriate for Method 2, because it has overlapping regions. The second has no overlapping regions, so it can be used for either Method 2 or Method 3.

### **Algorithm A**

For every vector a scan region is developed around the vector in order to cover all potential error points associated with the corresponding edge in the object. If the object is scanned with the sensor oriented parallel to the vector, and therefore to the edge, the error should be minimised. For parts where there are no scan regions the orientation of the scanner should not affect the result, so it can be chosen to be the default value, or some other if it is more convenient. Figure 8(a) – (e) illustrates this algorithm for the case of a simple L-shaped object. The width of the scan region needs to be large enough to cover all potential bow waves, because their extent is always greater than that for the spike distortions. Thus the width will need to be larger as the height of the object increases. The problem with this approach is that there are often overlaps between two or more scan regions and then the conflict between the different orientations needs to be resolved. We have taken a simple average of all the scan regions involved and, unfortunately, this can lead to errors. A better approach would be to use another algorithm for these problem areas, such as Algorithm B or even Wong’s method. Both these algorithms take longer but the additional time would not be very large in many cases, provided that the scan region widths are made as small as possible.

### **Algorithm B**

This algorithm uses the ‘nearest vector’ approach, whereby it is assumed that the vector (and therefore the edge) most likely to affect a point is the one nearest to that point. Thus there are no overlaps but the time taken to develop the regions is much greater. Figure 8(f) illustrates this algorithm for the simple L-shaped object from Figure 8(a) with vectors shown in Figure 8(b). In order to reduce the time taken, we have developed the idea of ‘regions of influence’, so that vectors which are not within a region of influence associated with a point can be assumed to have no influence on the errors at that point. The way we have done this means that some scan regions encroach into the default regions but that will not have a detrimental effect. The jagged edges of these encroachments can be seen in Figure 12(d).

## **7. Calibration**

In order to calibrate the image space against the scan space calibration markers (small circular disks painted with a black and white quarter pattern) were placed at known coordinates on the laser scan bed around the object to be scanned, using the laser itself as a guide to positioning the markers. The markers were placed around the object, then the camera was positioned over the object and the height

adjusted until all the calibration markers were visible in the camera's field of view. Once the coordinates have been found for these known points in the image space, the parameters for an affine transformation can be calculated, which can then be used to transform between the two spaces (Foley, Van Dam, Feiner, Hughes, & Phillips, 1994).

The camera needs to be correctly aligned, so that it is facing the scan bed perpendicularly, in order to minimise parallax errors in position of edges. This can be validated by moving the camera up and down and checking that a mark at the centre of the image remains at the centre, and correcting if necessary. For objects which are more than 3-4 cm. across, a correction will also be needed to allow for the parallax effect. Again this can be done by measuring the change in position of each edge as the camera moves up and down. This also opens up the possibility of automatic self-calibration of the system.

It is important that the laser scanner and sensor are set up such that the beam is aligned as closely as possible with the  $z$  axis of the CNC machine. If it is far from vertical, then some of the errors we are trying to avoid will be exacerbated and will be harder to avoid. The direction of the laser beam should be close to vertical but a small discrepancy can be compensated, if both the discrepancy in angle and its orientation are known. All three of the corrected coordinates of the measured point will generally be slightly different from the values read in.

It is also important that the scanner rotation mechanism is aligned as closely as possible with the  $z$  axis of the machine. Again a small discrepancy can be compensated, provided that its parameters are known. If the emitter is not on the axis of rotation, then the spot will move in a circle as the scanner is rotated. For Method 2 this can be compensated, once the radius is known. However, for Method 3 it would be difficult to accommodate more than a very small radius, because the measured point would be too far away from the intended position.

## 8. Results

Figure 1 shows the result obtained for a single scan of the small bottle top with the sensor oriented parallel to the  $x$  axis. No distortions are seen when the direction of the edge is roughly parallel to the  $x$  axis. But there are distortions when the edge is close to the direction of the  $y$  axis: 'spikes' at the top on the left and 'bow wave' near the base on the right.

Figure 10 shows the results obtained for a domino with slightly rounded corners. In Figure 10(b) and (c) the edge detection, vectorisation and regions obtained by Algorithm A look promising. Unfortunately the scan of the top of the domino (Figure 11) shows several downward spikes. This has occurred because of the low reflectivity of the black domino. Therefore we sprayed the domino with white powder and repeated the experiments as shown in Figure 10(d) – (f). The regions obtained for this by Algorithm A are not shown, because the overlaps make it confusing to interpret, but those for Algorithm B show how the scan regions relate to the vectors. The combined scans using both Algorithms A and B are shown in Figure 13(a) and (b). Algorithm B appears to give better results than Algorithm A in the overlapping parts where the averaging is used in Algorithm A.

Figure 12 shows the results for the small bottle top which has been sprayed. Again only the regions obtained for this with Algorithm B are shown, because those for A are confusing. Figure 1 shows the results for a single orientation scan with distortions present and Figure 14 shows the combined scan results. In Figure 14(a) the detail shows that with Algorithm A some downward spikes occur. However with Algorithm B, Figure 14(b) and (c), there are no significant distortions.

## 9. Discussion

From the results shown it appears that Algorithm B is better than Algorithm A, regardless of whether Method 2 or 3 is used. However, we have found that Algorithm B takes considerably longer for the software to run. Therefore a compromise method to find the regions may be the best approach. We propose to use Algorithm A initially but then use the output from it to identify all overlapping regions. For these regions Algorithm B can be used to resolve the conflict between several orientations, thus reducing the extra time required.

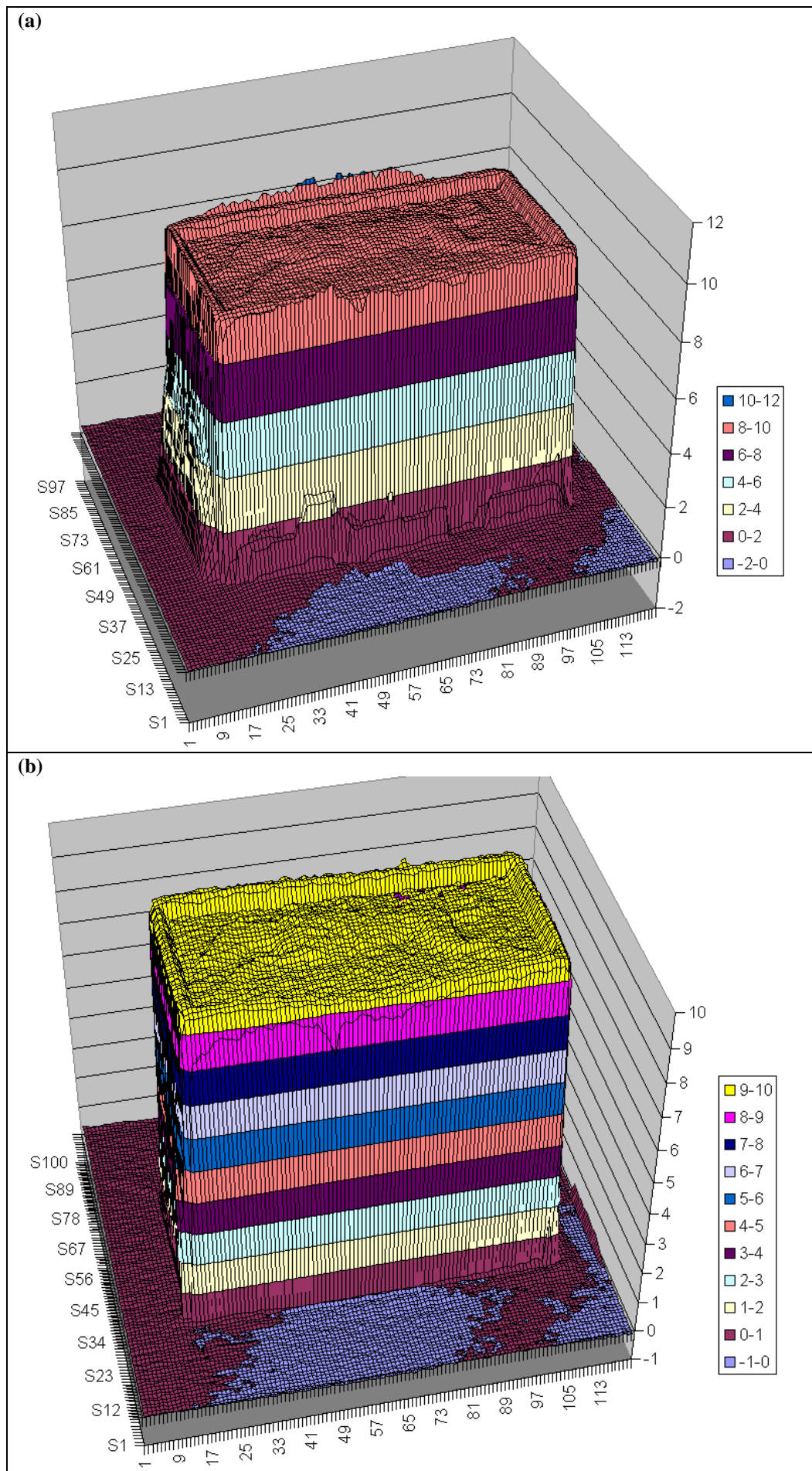


Figure 13: Combined scan results for the sprayed domino from Figure 10 using (a) Algorithm A and (b) Algorithm B. Again the scale of the vertical axis is exaggerated.



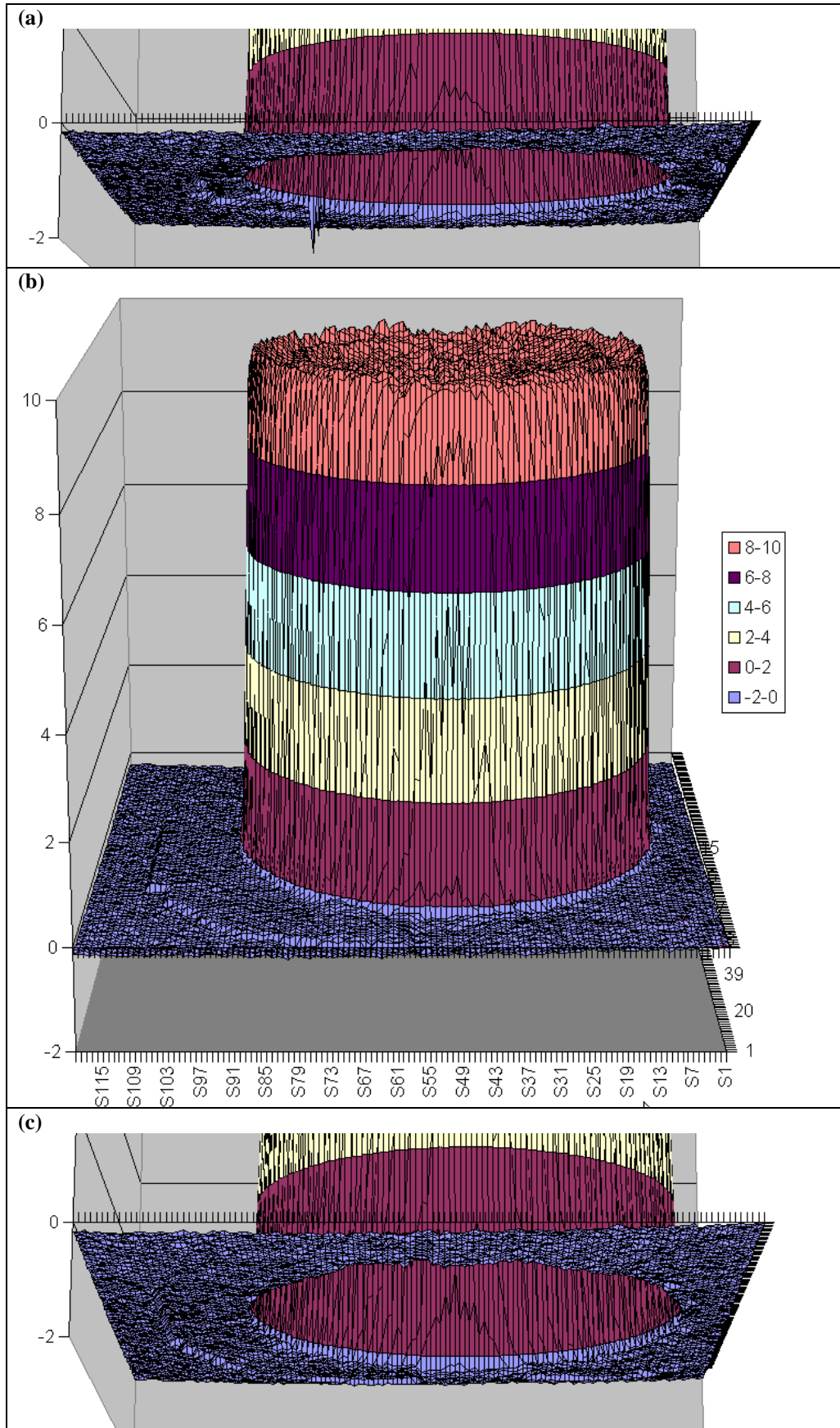


Figure 14: Scan results for the sprayed bottle top from Figures 1 and 12: (a) using Algorithm A, showing a detail viewed from below with a downward spike still present; (b) and (c) using Algorithm B, with views from above and a detail from below, showing no downward spikes. (The scale of the vertical axis is exaggerated.)



Another approach to reducing the time taken is to use Method 1 to identify the ‘best’ orientation and start performing a complete scan with this orientation. While this scan is taking place, the rest of the software can be run in parallel to plan the paths for the other orientations for Method 2.

There are a number of situations where it is not possible to resolve the problem of errors, because scans of all orientations will be distorted. Such situations occur when there is a small hole or an internal corner, as shown in orange in Figure 8(e) and no orientation can be relied upon to give undistorted results. For a narrow hole it may be impossible to detect light returning from the inside with any scanner orientation, which is a problem inherent in the nature of the sensor. Then all that can be done is to highlight such a problem region to the user. It may be possible to use extrapolation to try to predict the distortion near an internal corner, based on the way distortions occur further away from the corner. However, this relies on some assumption of homogeneity of the object. If such results are likely to be unreliable, those regions should also be highlighted to the user.

Other approaches could be used to reduce the total scan time. For example, it may be possible to estimate the heights of edges prior to scanning by using two images taken with the camera at different heights. The change in position will depend on the height of the edge and allow scan region width to be chosen appropriately. In addition the direction of movement will indicate which side is the lower, and therefore where the distortions are expected at some orientations. The scan region width for the other (higher) side can then be made much smaller.

Image analysis prior to scanning may also allow us to vary the scan resolution based on the complexity of the image. In areas of ‘high complexity’ we can increase the sample rate to achieve a better resolution where a higher level of detail is required.

## **10. Conclusions**

At the start of this Chapter, we specified that the replicating an existing solid object via CAD/CAM requires a cost effective method to generate data for input into the CAM system from that solid object. We have shown here that the integration of an optical camera into a single-perspective laser scanner system provides a major step towards such a method. The availability of reasonably inexpensive CCD cameras of reasonable pixel resolution makes this a cost effective solution to the distortion problems faced by such a scanner. Intelligent algorithms based on the fusion of the two forms of sensor output can give high quality results at a reduced cost when compared to more expensive laser scanners. Image analysis techniques allow the problem regions of the object to be identified, so that the path and orientation of the laser sensor can be planned before the scan, thereby reducing distortions.

Care needs to be taken with both the setting up and calibration of the camera relative to the scanner and of the scanner rotation system, in order to ensure that other errors are not introduced. However, small misalignments can be detected and compensated in the software system. Work is in progress in our laboratory towards integrating such a system into operational equipment.

## **11. Acknowledgements**

The support of the EPSRC and Axiomatic Technology Ltd. in funding and facilities for this project is gratefully acknowledged.

## **12. Future Research Directions**

This chapter has outlined work which represents the first step in implementing the use of our inexpensive and cost effective approach to scanning 3D objects, so that they can be replicated by conventional CAD/CAM techniques. There remains a significant body of research and development which will be needed before our approach can be implemented as a routine technique in mechanical engineering workshops and similar industrial environments. The first stage involves testing our approach on a wide range of objects, including those which contain holes and concave features. There is no reason to suppose that this method will not work with such features, except where occlusion of the beam by another part of the object is impossible to avoid from any orientation. However, it is important to validate the technique with real examples. The second stage involves designing a robust scanner for carrying out these measurements, as opposed to the modified laboratory instrument that was

used in our studies described above. Given the increasing move towards CAD/CAM technology and the decline in the number of engineers trained in traditional machining techniques, we believe that our approach will become increasingly viable in years to come.

### 13. References

- Ballard, D.H. (1981). Generalizing the Hough Transform to Detect Arbitrary Shapes. *Pattern Recognition*, 13 (2) 111-122.
- Canny, J.F.A. (1986). Computational Approach to Edge Detection. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 8 (6), 679-698.
- Chen, P.C. & Pavlidis, T. (1981). Image Segmentation as an Estimation Problem. In A. Rosenfeld (Ed.), *Image Modelling* (pp. 9–28). Academic Press.
- Chow, Y.K., Poliakoff, J.F. & Thomas, P. D. (2002). Interpolation and Acceleration Algorithms for Stepper Motors – a Parametric Approach. In *Proceedings of 8<sup>th</sup> IEEE International Conference on Methods, Models and Robotics*, ISBN 83-88764-51-9.
- Denby, A., Langensiepen, C., Poliakoff, J. & Sherkat, N. (2005). Use of Edge Models to Guide Optimal 3D Scanning,. In *Proceedings of 8<sup>th</sup> International Conference on Computer Modelling and Simulation*, ISBN: 1-84233-111-6.
- Deriche, R. (1987). Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector. *The International Journal of Computer Vision*, 1 (2), 167-187.
- Foley, J.D. Van Dam, A., Feiner, S.K., Hughes, J.F., & Phillips, R.L. (1994). *Introduction to Computer Graphics*. (pp.168-177), Addison-Wesley, MA.
- Freeman, H. (1970). Boundary encoding and processing. In *Picture Processing and Psychopictories*. New York: Academic Press.
- Haralick, R. Shanmugam, K. & Dinstein I. (1973). Texture Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC- 3 (6) 610-621.
- Heath, M., Sarkar, S., Sanocki, T. & Bowyer, K. (1998). Comparison of Edge Detectors, A Methodology and Initial Study. *Computer Vision and Image Understanding*, 69 (1), 38-54.
- Hori, O. & Tanigawa, S. (1993). Raster-to Vector Conversion by Line Fitting Based on Contours and Skeletons. In *Proceedings of ICDAR'93*, (pp. 353-358). Tsukuba, Japan.
- Ji, Q. & Xie, Y. (2003). Randomised Hough Transform with Error Propagation for Line and Circle Detection. *Pattern Analysis Applications* 6, 55-64.
- Kitchen, L. & Rosenfeld, A. (1982). Gray-level Corner Detection. *Pattern Recognition Letters*, 1 95–102.
- Low, A. (1991). *Introductory Computer Vision and Image Processing*. (p.69) McGraw-Hill.
- Lyvers, E.P. & Mitchell, O.R. (1988). Precision Edge Contrast and Orientation Estimation. *IEEE. Transactions on Pattern Analysis and Machine Intelligence*, 10 (6), 927-937.
- Novak, C.L. & Shafer, S.A. (1987). Color Edge Detection. In *Proceedings DARPA Image Understanding Workshop: Vol. I*. (pp. 35-37). Los Angeles, CA.
- Poliakoff, J.F. & Thomas, P.D. (1998). Error Correction in Scanned Engineering Drawings using 3-D Knowledge-Based Reconstruction. In R. Kasturi, A.K. Chhabra (Eds.) *Graphics Recognition*, Lecture Notes in Computer Science, Vol. 1389, Springer-Verlag, (pp. 280-290) ISBN 3 540 64381 8.
- Poliakoff, J.F., Chow, Y.K., Orton, P.A., Howson, M. & Al-Dabass, D. (2005). Evaluation by Simulation of Interpolation and Acceleration Algorithms for Stepper Motors. *International Journal of Simulation Systems, Science & Technology*, 6 (7-8), ISSN 1473-804x (online), 18-29.
- Prewitt, J.M.S. (1970). Object Enhancement and Extraction. In B. S. Lipkin and A. Rosenfeld (Eds.), *Picture Processing and Psychopictories*, New York: Academic Press.
- Risse, T. (1989). Hough Transform for Line Recognition: Complexity of Evidence Accumulation and Cluster Detection. *CVGIP*, 46 (3) 327-345.
- Sacchi, R., Poliakoff, J.F., Thomas, P.D. & Häfele, K.-H. (2004). Segmentation of Scanned Surfaces: Improved Extraction of Planes. In M. Sarfraz (Ed.), *Geometric Modeling: Techniques, Applications, Systems and Tools*, (pp. 251-262). Wiley.
- Smith, S.M. & Brady, J.M. (1997). SUSAN - a new approach to low level image processing, *International Journal of Computer Vision*, 23 (1), 45–78.
- Sezgin, M. & Sankur, B. (2004). Survey Over Image Thresholding and Quantitative Performance Evaluation. *Journal of Electronic Imaging*, 13 (1) 146-165.
- Spann, M. & Wilson, R.G. (1985). A Quad-Tree Approach to Image Segmentation Which Combines Statistical and Spatial Information. *Pattern Recognition*, 18 (3/4) 257–269.

- Tombre, K. (1998). Analysis of engineering drawing: state of the art and challenges. In K. Tombre, & A.K. Chhabra (Eds), *Graphics Recognition - Algorithms and Systems* (pp. 257-264). Berlin: Springer-Verlag.
- Tombre, K. & Tabbone, S. (2000). Vectorization in Graphics Recognition: To Thin or Not To Thin. In *Proceedings of ICPR'00: Vol.* (pp. 291-96).
- Wang, S.J. & Binford T.O. (1994). *Generic, Model-based Estimation and Detection of Discontinuities in Image Surfaces*. Volume II, (pp.113-116). ARPA IUW.
- Wong, K.H., Poliakoff, J.F. & Thomas, P.D. (2001). Compensation Techniques for Distortions from a Single-Perspective Optical Triangulation Sensor. In *Proceedings of Fifth Conference on Optical 3-D Measurement Techniques*, ISBN 3-9501492-0-1.
- Wong, K.H. (2002) *Compensation for Distortion in the Imaging Process for 3D Surfaces* (PhD Thesis). Nottingham, UK: Nottingham Trent University.
- Ziou, D. & Tabbone, S. (1998). Edge Detection Techniques - An Overview. *Pattern Recognition and Image Analysis*, 8 (4), 537-559.

## 14. Additional Reading

- Beraldin, J.A., Blais, F., Cournoyer, L., Rioux, M., El-Hakim, S.H., Rodella, R., Bernier, F. & Harrison, N. (1999). 3D Digital Imaging and Modeling on Remote Sites. In *Proceedings Second International Conference on 3D Digital Imaging and Modelling*, Ottawa, Canada.
- Blais, F. (2004). Review of 20 Years of Range Sensor Development. *Journal of Electronic Imaging* 13 (1). 231-240.
- Chai, I. & Dori, D. (1992). Orthogonal Zig-Zag: An Efficient Method for Extracting Lines From Engineering Drawings. In C. Arcelli, L.P. Cordella & G. Saruiti di Baja (Eds.), *Visual Form*. New York: Plenum Press, 127-136.
- Chen, F., Brown, G.M. & Song, M. (2000). Overview of Three-Dimensional Shape Measurement using Optical Methods. *Optical Engineering*, 39. 10-22.
- Curless, B. (1997). *New Methods for Surface Reconstruction from Range Images*. (PhD Thesis). USA: Stanford University.
- De Bakker M., Verbeek P.W. & Steenvoorden G. K. (2000). Smart PSD Array for Sheet-of-Light Range Imaging. In *Proceedings of SPIE*, 3965, 21-32.
- Dorsch R., Herrmann J. & Häusler G. (1994). Laser Triangulation: Fundamental Uncertainty of Distance Measurement. *Applied Optics*, 33 (7) 1306-1314.
- Faugeras, O.D., Luong, Q.-T. & Maybank, S.J. (1992). Camera Self-Calibration: Theory and Experiments. In *Proceedings of European Conference on Computer Vision*, 321-34.
- Faugeras, F. (1993). *Three-Dimensional Computer Vision*, USA: MIT Press.
- Häusler G. (1999). Three-Dimensional Sensors – Potentials and Limitations, In *Handbook of Computer Vision and Applications*, Academic Press, 1 (19), 485-506.
- Ingle K.A. (1994). *Reverse Engineering*, McGraw-Hill Professional Publishing.
- Jansa J., Huang Y. R. & Trinder J. C. (1993). Problems of Precise Target Location and Camera Orientation in Digital Close-Range Photogrammetry. In *Proceedings of SPIE, Videometrics II*, 2067, 151-161.
- Kanaugo T., Jaisimha, M., Palmer J. & Haralick, R. (1995). A Methodology for Quantitative Performance Evaluation of Edge Detection Algorithms, *IEEE Transactions on Image Processing*, 4 (12) 1667-1673.
- Nalwa, V.S. (1993). *A Guided Tour of Computer Vision*, Addison-Wesley.
- Reichmann W. (1995). Fast Object Recording by means of Structured Light and Photogrammetric Techniques, In *Proceedings of IAPRS*, 30, 195-200.
- Ruzon M. & Tomasi, C. (1999). Color Edge Detection with the Compass Operator, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Ft. Collins, CO, 2, 160-166.
- Shu C. & Xi F. (1999). Model-Based Scanning Path Generation for Inspection, In *Proceedings of Second International Conference on 3-D Digital Imaging and Modeling (3DIM '99)*, Ottawa, Canada, 118-124.
- Thomas, P.D., Poliakoff, J.F., Razzaq, S.M. & Whitrow, R.J. (1996). A Combined High and Low Level Approach to Interpreting Scanned Engineering Drawings. In R. Kasturi & K. Tombre (Eds.) *Graphics Recognition - Methods and Applications*, Lecture Notes in Computer Science, Vol. 1072, Springer-Verlag.
- Vàrady T., Martin R.R. & Cox J. (1997). Reverse Engineering of Geometric Models – An Introduction. *Computer Aided Design*, 29, (4) 255-268.
- Xu Y., Tang J. & Zhong W. (2000). New Method of Processing the Signals of a Position Sensitive Detector. In *Proceedings of SPIE, Advanced Photonic Sensors: Technology and Applications*, 4220, 260-263.
- Zheng, Z., Wang, H. & Teoh, E. (1999). Analysis of Gray Level Corner Detection, *Pattern Recognition Letters*, 20, 149-162.

## Appendix D

This appendix presents the Paper submitted for the UKSIM'05 Conference at St. John's College, Oxford. A presentation of the work was also given at the conference.

A. Denby, C. Langensiepen, J. Poliakoff, N. Sherkat, "Use of Edge Models to Guide Optimal 3D Scanning", *Proc. 8<sup>th</sup> ICCMS (8<sup>th</sup> International Conference on Computer Modelling and Simulation*, ISBN: 1-84233-111-6, Oxford, UK, 6 - 8 April 2005.

# USE OF EDGE MODELS TO GUIDE OPTIMAL 3D SCANNING

A DENBY, C. LANGENSIEPEN, J. POLIAKOFF, N. SHERKAT

School of computing & informatics  
Nottingham Trent University  
Nottingham, NG1 4BU, UK.  
e-mail: janet.poliakoff@ntu.ac.uk

**Abstract:** laser scanning of objects for reverse engineering is subject to distortions. These false readings are often generated when the laser head is oriented perpendicular to an edge of the object and the laser beam is occluded or reflected. The aim of this work is to reduce the occurrence of such distortions by preplanning the scanning process to avoid such orientations, allowing an accurate scan in a reasonable elapsed time. By applying edge detection to a digital image of the object, a model of the edges can be generated, and the scan path mapped to ensure correct orientation. It has been found that simple edge detection algorithms such as canny allow a single best orientation to be determined, but to build a model of the scanning process requires a combination of algorithms to eliminate noise and create continuous edge segments.

## INTRODUCTION

In traditional 'forward' engineering concepts and models are transformed into real parts. In reverse engineering real parts or prototypes are transformed into engineering models and concepts. Typically, reverse engineering begins by measuring an existing object so that a model can be deduced in order to exploit the advantages of CAD/CAM technologies to construct a new, improved product. The data acquisition phase is a crucial step in this procedure.

Laser triangulation is a popular non-contact data acquisition method in which a laser beam is projected onto the surface of interest and the reflected spot is detected by one or more photosensitive devices. Laser triangulation can acquire data at very high rates; however the technique is subject to a number of different errors. Previous work by Wong [1] has identified that the majority of these errors fall into three broad categories: (systemic) noise, transitional errors caused by changes in reflectivity across the object and errors due to the geometry of the object. In some cases the reflected signal from the illuminated spot is occluded from the sensor by another part of the object and stray light may cause a large distortion in the reading. In other cases the spot is not occluded but secondary reflections cause a smaller distortion, because two spot are somehow 'averaged'. Wong found that errors in this third geometric category are at their worst when the laser scan head is oriented perpendicular to the edge of the object being scanned (see Fig.1). Conversely, if the laser is oriented parallel to the 'edges' of the object then errors are minimised (as shown in Fig.2). As found by Wong, the problem can be

much reduced by performing multiple scans at a range of orientations. His algorithms then combined the resultant data by comparing the data from different orientations at a particular location and deducing which were the least distorted values and discarding the rest. However, such full multiple scans require an unacceptable length of time, because even a single scan of 0.05 mm pitch requires about 1 hour to execute for an object of 30 sq cm. Our aim is to achieve the reduction in errors by detecting the optimal orientations prior to scanning, thus significantly reducing the total scan time required.

## NEW IDEAS

Our hypothesis is that recognition of problem areas prior to the laser scan process can be used to determine an optimal scanning process. Planning of the scanning strategy will involve selecting the optimal orientation of the sensor unit for different regions of the object, thereby minimising the 'geometric' errors described earlier. An obvious way to do this would be to build a simple interactive program that requires the operator to identify the 'important' edges on an image of the object, and then uses this information to generate the scanning strategy. This would take advantage of the human skill which allows us to 'see' lines and directions even when the image is complex, or has a noisy background. The image of the object would have to be generated via a digital camera, presented on a standard VDU and then the edges selected via a mouse. Some form of calibration would also have to be undertaken to match the image to the scan positions.

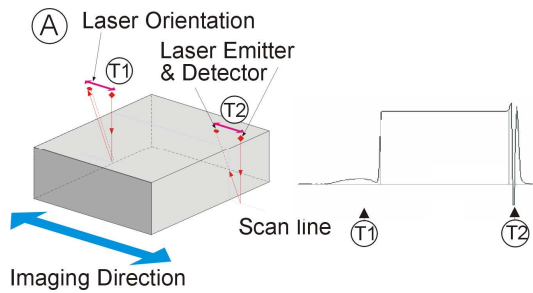


Figure 1 Occlusion causing scan artefacts when scan head oriented perpendicular to edge

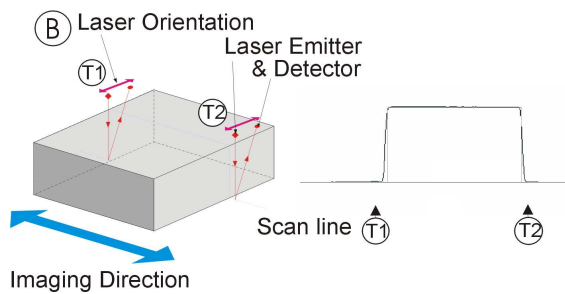


Figure 2 Reduced occlusion with changed orientation

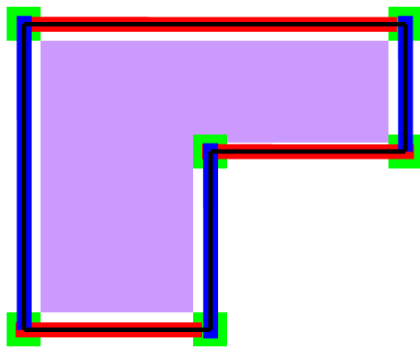


Figure 3 – for this shape, lilac indicates where head orientation is non-critical, red/blue show where orientation must be aligned with X/Y axes, and green where samples must be combined from multiple partial scans

The problem with such a method is that it requires considerably greater operator skills and even then the operator can make mistakes – perhaps ignoring shorter edges even though the sharpness of that edge could give rise to significant errors in the point cloud. Moreover, complex parts could require a significant quantity of operator time in identifying key edges and selecting scan regions. This would be commercially impractical.

Our approach is to automate the above process by simulating this human skill of abstraction, by using

directly information captured from a simple digital camera mounted above the scanning bed. The camera image is analysed using edge detection techniques, and a scanning path that minimises errors due to object geometry can then be generated based on the edge map.

The final stage will be deriving scan regions that form broad bands around the edges, so that the any offset errors in the calibration can be accounted for, and so that there are sufficient scan lines at the required orientation to enable smoothing of the results. The overall process would then involve a digital image, followed by a full scan. While the scan was taking place, the above processing could be performed, resulting in the list of bands enclosing edges where the geometric errors might occur. On completion of the full scan, the scan head can be realigned, and only the bands relevant to that orientation would then be rescanned. This would be repeated for each head orientation, resulting in a suite of results that can be merged and smoothed over the edges to provide a more accurate point cloud in a reasonable time.

The core of this work involves evaluating and combining algorithms to achieve the best combination for this particular purpose. We consider the problem to have the following requirements and constraints:

- In order to give a commercial benefit in both quality and time, the final system would have to perform the whole process in a time not much greater than that of a single scan.
- The algorithms must have good noise removal, as noise can manifest itself as small false edge segments that will also extend the overall scan time.
- The algorithms should provide information about the edges – not just their positions and extents, but directional information, since the scanner has to be appropriately aligned
- To minimise user intervention and reduce the skill level required of the operator, the image processing should not be very sensitive to parametric changes. (Once the scanner has been set up, the same thresholds should be applicable to a wide range of scanned objects.)
- Calibration should be as simple as possible, preferably using a predefined image on the scan bed to align the laser and camera.

## EDGE DETECTION ALGORITHMS

### First Derivative (Gradient) Edge Detectors

The magnitude of the first derivative can be used to detect the presence of an edge in an image. This is obtained by performing a matrix convolution of the image with a filter or ‘mask’. Ideally, small-sized masks are employed in order to detect fine variation in grey level distribution (i.e. micro-edges). On the other hand, large-sized masks are required in order to detect coarse variation in grey level distribution (i.e. macro-edges) and to filter-out noise and other irregularities. The Roberts Cross, Sobel, Prewitt gradient method, Kirsch and Prewitt compass operators and the Frei-Chen method as examples of first derivative operators. [2]

The effectiveness of the Canny operator is determined by three parameters - the standard deviation of the Gaussian used in the smoothing phase and the upper and lower thresholds used by the edge-tracking. Increasing the width of the Gaussian kernel reduces sensitivity to noise at the expense of losing some of the finer detail in the image. The localisation error of detected edges also increases as the Gaussian width is increased.

In most cases the upper tracking threshold can be set quite high and the lower threshold quite low with a good tolerance in the results. Setting the lower threshold too high causes noisy edges to break up. Setting the upper threshold too low increases the number of spurious ‘noise’ edges appearing in the output.

An algorithm based on the Canny edge detector has been developed and evaluated employing a Gaussian smoothing function, simple 1st derivative masks for detection and a recursive edge hysteresis technique. This method uses both the magnitude and orientation values for the hysteresis process in order to provide directional edge following and assigns an ‘orientation value’ to the edge. [3]

### Second Derivative Detectors

Second order derivatives can be used to detect the position of maximum change in the gradient. The Laplacian function provides a measure of the 2nd spatial derivative of an image however they are highly sensitive to noise in the image. Second derivative masks can be combined with Gaussian smoothing (i.e. Laplacian of Gaussian or LoG) in order to reduce their sensitivity to noise. The zero crossings of the second derivative provide a useful way of locating edges in an image. The sign of the second derivative can be used to determine whether an edge pixel lies on the dark or light side of an edge. [2]

Second derivative operators are isotropic, and as such do not provide any information about edge orientation and form closed loop ‘contours (except where the edge extends beyond the image area) – this leads to what is commonly called the ‘plate of spaghetti’ effect where the confusion of loops detracts from the appearance of detected edges. The Hough Transform can be used to provide extra information for the extraction of edges. The main problem with the large number of pixels used to provide evidence of lines is the well-known ‘bow tie’ or ‘butterfly’ effect associated with the Hough transform. By using a ‘windowed’ multiscale Hough transform, in which a small window is passed over the image and only the locally detected edges are included in the transform, this effect is reduced and the detected Hough Lines are more localised to the edges. [4-6]

## RESULTS

The results of our evaluations are shown with reference to a sample object – a cookie cutter with distinct vertical edges, of approx. 30 sq cm area. The cutter was placed on a white background for imaging, though fig 4a shows that the angle of the light caused some shadows – this was not corrected as it proved a useful additional test for the algorithms. The image being processed was monochrome and 640 by 480 pixels. Table 1-3 show the relative times taken to perform the algorithms, while figure 4b -4h show the resultant edges.

**Table 1 Typical execution times for edge algorithms**

Method	Mask Size	No. Masks	Mean time (ms) (for 5 samples)
Robert's Cross	2x2	2	173.5
Prewitt Gradient	3x3	2	275.3
Sobel	3x3	2	273.2
Frei-Chen	3x3	9	1047.5
Extended Prewitt	5x5	2	545.6
Canny(convolution only)	1x3	2	155.7

**Table 2 Typical execution times for compass algorithms**

Method	Mean time (ms) for magnitude calc. (N=5)	Mean time (ms) for orientation calc. (N=5)	Mean time (ms) for operator +NMS + edge hysteresis thresholding (N=5)
Prewitt gradient (sum abs. values)	231.9	237.5	755.2
Prewitt compass	778.7	826.9	1806.3
Kirsch compass	791.9	804.9	1801.6

**Table 3 Typical execution times for second derivative techniques**

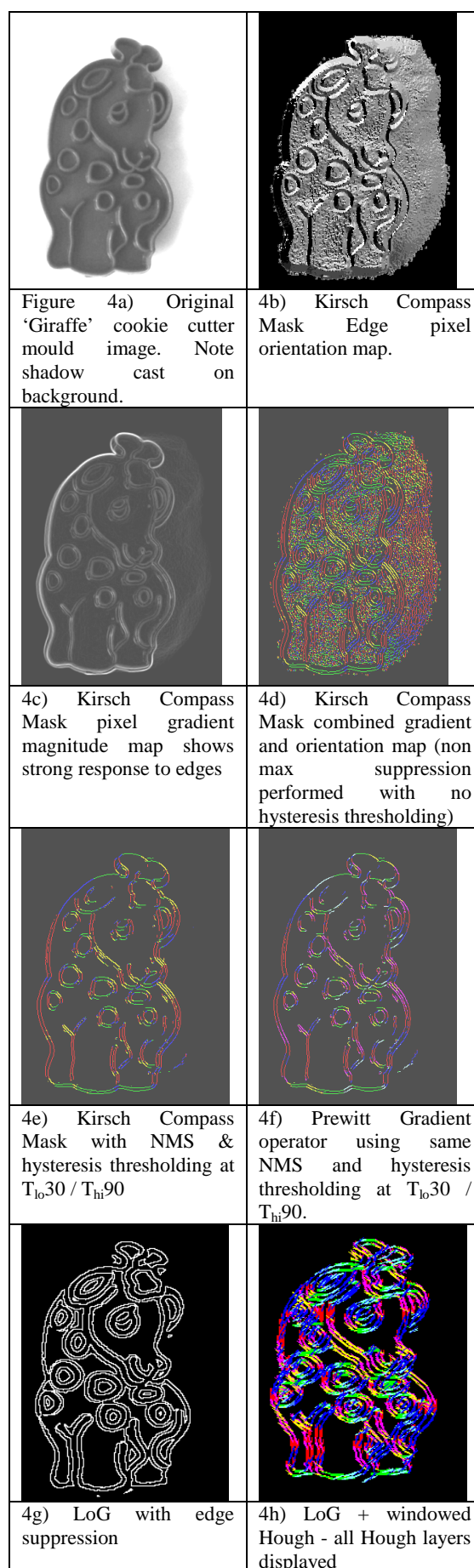
Method	Mean time (Seconds)
LoG with sigma 1.4	2.94
LoG + Windowed	398
Hough	

As can be seen from the images and times, Roberts, Prewitt and Sobel are very fast. The Canny method is also a very effective edge detector, but it and the other gradient methods require thresholds to be set on an image by image basis in order to obtain optimum results. This is undesirable – as stated earlier, we wish to minimise operator interaction – so some additional processing or alternative algorithm would be preferable.

The 2<sup>nd</sup> derivative methods proved to be quite effective when applied as zero-crossing of LoG. Noise edge suppression became satisfactory when using a threshold for the 3<sup>rd</sup> derivative based on a combination of global and local variance rather than the normal overall image variance. They also appeared to be less sensitive to parameter values than the gradient methods such as Canny. This is important in reducing the need for skilled operator intervention. However, applying the Windowed Hough technique to the resultant edges in order to add orientation information took a significant amount of processing time – though still a factor of 10-15 shorter than a typical scan time.

### Initial Solution

An initial solution with some improvement on the basic scan has been completed. The requirement to minimise operator intervention was relaxed, and it was assumed that the operator could be asked to perform a short and simple task. An algorithm based on the Canny edge-following output has been developed to approximate the optimal orientation for a single overall scan across the whole image space. A simple count of the number of edge pixels assigned to each defined orientation ‘bucket’ was performed. The bucket with the highest pixel count was chosen as representative of the best orientation and the operator prompted to rotate the object under the camera by a recommended angle in order to align the object with the scanner. Although, as stated above, optimum performance requires ‘tuning’ of the threshold parameters, a standard set were used here as the requirement was only to find the most frequently observed direction, and false positives or missing small lines were not critical.





## Calibration

In order to ensure that the scan head is oriented correctly, we cannot simply assume that the scan direction is parallel to the camera image axis. We also need to know the size and position of the object relative to the scan start position so that scanning can be limited to the object area only. It is therefore necessary to carry out calibration. A standardised printed card of known dimensions is positioned centrally on the scanner bed. Edge detection can then be used to determine the dominant orientation of this image (thus recording the relative angle between scanner and camera), while the offset from the centre of the image gives the camera offset relative to the scanner head. Given a camera height of approx. 30 cm from the scanner bed, and a firm mounting, it has been found that lens distortion and non-vertical image axis do not appear to have significant effect on the calibration.

## Full Solution

Our investigation of the available algorithms now leads to two potential solutions. We may combine a fast gradient method with another technique to determine detailed angular information. However, we may risk missing some edges due to poor choice of threshold parameters. The technique currently under investigation generates accurate angular information involves 'growing' edge fragments by coalescing them with nearby fragments of similar orientation, and then deriving the actual edge orientation from the x,y positions of the new extended ends. The decision as to whether to include a new fragment depends on the current angle deduced for the existing edge, the distance of the fragment, its quantised direction, and magnitude.

Another method is to use a 2<sup>nd</sup> derivative method coupled with the windowed Hough technique to derive good angular and magnitude information. This has been found to result in sets of edges bucketed at the desired angles, though not coalesced into contiguous scan regions. This method is relatively slow due to the Hough processing, but still fast compared to a scan.

Although it might appear that we could combine the threshold insensitivity of the 2<sup>nd</sup> derivative methods and the single line production of the gradient methods to produce a combined result for edge growing. However, the process is not straightforward. The positions of the maximal values for the 1<sup>st</sup> and 2<sup>nd</sup> derivatives for the edges are not always coincident on the image, and the relative position differs for individual edges. Thus deriving a common edge map requires additional processing.

## CONCLUSIONS AND FUTURE WORK

While the initial solution proved to be a satisfactory initial improvement to the process, we are now in the process of developing and evaluating the two alternatives for the full solution. Initial work on the second derivative solution suggests that a considerable reduction in time to achieve an accurate scan is possible, though further measurements and timings need to be taken to assess the actual improvement in accuracy and time for a range of typical objects. It appears that, with additional work on the algorithms, accuracy could be improved still further by the appropriate conditions of the partial scans.

Future work would include the use of the optical images, possibly with directional lighting to give extra information from shadows, to establish other locations for scan error due to occlusion, and overcome them.

## Acknowledgements

We gratefully acknowledge the support of the EPSRC and Axiomatic Technology Ltd in carrying out this project.

## References

1. K.H. Wong, 'Compensation for Distortion in the Imaging Process for 3D Surfaces', PhD. Thesis, The Nottingham Trent University, 2002.
2. Ziou D., Tabbone S., "Edge Detection Techniques: An Overview", Pattern Recognition and Image Analysis, vol. 8, no. 4, 1998
3. J. F. Canny, 'A Computational Approach to Edge Detection', IEEE Trans. Pattern Analysis & Machine Intelligence, Vol. 8, No. 6, 1986. pp. 679-698.
4. K. Hansen, and J. Damgaard Andersen, 'Understanding the Hough Transform: Hough cell support and its utilisation' Image and Vision Computing 15 (1997) pp. 205-218
5. Duda, R.O., Hart, P.E. 'Use of the Hough transformation to detect lines and curves in pictures' Comm. Of the ACM. Vol. 15, 1972 No. 1 pp. 11-15
6. Denby, J Poliakoff, C. Langensiepen, N. Sherkat, "Integration of Optical Information for Accurate Representation of 3 Dimensional Surfaces in Computer-Aided Manufacturing", Fourth Conference on Postgraduate Research in Electronics, Photonics, Communications and Software (PREP2004), April 2004